



Tribhuvan University

Institute of Science and Technology

# News Clustering System based on Text Mining

## A Dissertation

Submitted to

Central Department of Computer Science and Information Technology

Tribhuvan University, Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements

For the Master's Degree in Computer Science and Information Technology

Submitted by

**Deni Shahi**

September, 2016



Tribhuvan University

Institute of Science and Technology

# News Clustering System based on Text Mining

## **A Dissertation**

### **Submitted to**

Central Department of Computer Science and Information Technology

Tribhuvan University, Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements

For the Master's Degree in Computer Science and Information Technology

### **Supervisor**

**Prof. Dr. Shashidhar Ram Joshi**

### **Co-Supervisor**

**Bikash Balami**

### **Submitted by**

**Deni Shahi**

September, 2016



**Tribhuvan University**  
**Institute of Science and Technology**  
**Central Department of Computer Science and Information Technology**

**Student's Declaration**

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

.....

**Deni Shahi**

**Date:** 20 September, 2016



**Tribhuvan University**  
**Institute of Science and Technology**  
**Central Department of Computer Science and Information Technology**

**Supervisor's Recommendation**

I hereby recommend that this dissertation prepared under my supervision by **Ms. Deni Shahi** entitled “**News Clustering System based on Text Mining**” in partial fulfillment of the requirements for the degree of M. Sc. in Computer Science and Information Technology be processed for the evaluation.

.....

**Prof. Dr. Shashidhar Ram Joshi**

Department of Electronics and Computer Engineering,  
Institute of Engineering,  
Pulchowk, Nepal

**Date:** 20 September, 2016



**Tribhuvan University**  
**Institute of Science and Technology**  
**Central Department of Computer Science and Information Technology**

**LETTER OF APPROVAL**

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirements of Masters Degree in Computer Science and Information Technology.

**Evaluation Committee**

.....  
**Asst. Prof. Nawaraj Paudel**  
**Head of Department**  
Center Department of Computer Science  
and Information Technology,  
Tribhuvan University, Kirtipur, Nepal

.....  
**Prof. Dr. Shashidhar Ram Joshi**  
Department of Electronics and Computer  
Engineering  
Institute of Engineering,  
Pulchowk, Nepal

.....  
**(External Examiner)**

.....  
**(Internal Examiner)**

**Date:** 26 October, 2016

## Acknowledgement

Foremost, I would like to express my sincere gratitude to my supervisor **Prof. Dr. Shashidhar Ram Joshi**. I have been amazingly fortunate to have a supervisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. His patience and support helped me overcome many crisis situations and finish this dissertation.

Besides my supervisor, I would like to special thanks to my co-supervisor **Mr. Bikash Balami** who gave me the lots of ideas and support to complete this work.

I would like to thank the research committee for their encouragement, insightful comments, and hard questions. I am indebted to all the people who supported and encouraged me involving directly or indirectly to complete this work. I am also obliged to **Head of Department, Asst. Prof. Nawaraj Paudel** and all respected teachers and staffs of Central Department of Computer Science and Information Technology, Tribhuvan University for their cooperation to bring this work in a tangible form.

I am very much thankful to **Mr. Ashish Singh Bista** for his valuable time and effort to complete this work.

Last but not the least, I would like to thank my family for their love and supporting me spiritually throughout my life.

## Abstract

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. This dissertation entitled “**News Clustering System based on Text Mining**” is one of the implementation of Data Mining in which the similar type articles of different Newspapers are grouped together which is in English language.

In this work, documents from different newspapers’ sites are retrieved i.e. Information Extraction (IE) using crawler then document preprocessing is applied. Parser parses the data into article heading and corresponding links, then the headings are split into individual terms and a list of distinct terms are maintained. Then the porter stemming algorithm is applied over the distinct terms collection. Stemming minimizes the vocabulary size (i.e. no. of terms will be minimized). TF-IDF of individual heading is calculated. This process represents individual content and heading in to n-dimensional vector space (n is the number of distinct terms in the article). Finally, K-means algorithm is implemented to group the news.

The Efficiency of K-means Clustering Algorithm has been analyzed for different values of initial number of cluster seeds (K) and different iterations (I). The result analysis is on seven days news data. The result obtained by the experiment shows that the result is efficient with the initial clusters seed 12 (K=12), Iterations to maintain the constant cluster centers in K-means clustering depends upon the number of data sets and running time is also directly proportional to the number of iterations and number of initial clusters seeds.

*Keywords: Data Mining, Information Extraction, Document Preprocessing, Porter Stemming Algorithm, TF-IDF, K-means Clustering Algorithm*

# Contents

Abstract.....	i
List of Figures.....	iv
List of Tables.....	v
List of Abbreviations.....	vi
CHAPTER 1 .....	1
INTRODUCTION .....	1
1.1 THESIS ORGANIZATION.....	2
CHAPTER 2 .....	3
BACKGROUND STUDY AND PROBLEM FORMULATION .....	3
2.1. BACKGROUND .....	3
2.1.1 Information Extraction.....	3
2.1.2 Web Crawler .....	3
2.1.3 Document Preprocessing .....	4
2.1.4 Clustering.....	8
2.2 PROBLEM FORMULATION.....	11
2.2.1 Problem Statement.....	11
2.2.2 Objectives .....	12
2.3 MOTIVATION .....	12
CHAPTER 3 .....	14
LITERATURE REVIEW AND METHODOLOGY.....	14
3.1 RELATED WORKS.....	14
3.2 RESEARCH QUESTION.....	16
3.3 PROPOSED FRAMEWORK.....	17
3.3.1 News Content Extraction .....	17
3.3.2 Parsing.....	17
3.3.3 Document Preprocessing .....	19
3.3.4 Document Representation.....	19
3.3.5 Document Clustering .....	19
CHAPTER 4 .....	20
IMPLEMENTATION.....	20
4.1 TOOLS USED .....	20
4.1.1 Resource Requirements .....	20
4.1.2 Programming Language.....	20



4.2 DATA SOURCE MODULE.....	21
4.3 News Extraction Module .....	22
4.4 DOCUMENT PREPROCESSING MODULE.....	23
4.5 CLUSTERING MODULE.....	26
4.6 EVALUATION MODULE .....	29
CHAPTER 5 .....	31
DATA COLLECTION AND ANALYSIS.....	31
5.1 DATA COLLECTION .....	31
5.1.1 Sources.....	31
5.1.2 News Data.....	32
5.2 EXPERIMENTAL RESULT.....	36
5.2.1 Experimental Setup.....	36
5.2.2 Sample Output .....	36
5.2.3 Evaluation Metrics.....	38
5.2.4 Result .....	43
CHAPTER 6 .....	45
CONCLUSION AND FUTURE WORK .....	45
6.1 CONCLUSION.....	45
6.2 FUTURE ENHANCEMENTS .....	46
References.....	47
Bibliography .....	49
APPENDIX.....	50
Implementation Code.....	50
Evaluation Table.....	60

## List of Figures

Figure 2.1.2.1 Architecture of Web Crawler .....	4
Figure 2.1.2.2 Basic Architecture of Web Crawler.....	5
Figure 2.1.4.1.1 Representation of document in Vector Space.....	10
Figure 3.3.1 Proposed Framework for News Clustering.....	16
Figure 5.2.1.1 Screenshot of Sample News headings extracted from respective portal.....	33
Figure 5.2.1.2 Screenshot of Sample terms of news before preprocessing.....	34
Figure 5.2.1.3 Screenshot of Sample terms of news after preprocessing.....	34
Figure 5.2.2.1 Screenshot of Output of total News of each portal.....	35
Figure 5.2.2.2 Screenshot of Output News before clustering.....	36
Figure 5.2.2.3 Screenshot of Output News after Clustering.....	36
Figure 5.2.2.4 Screenshot of Output of Evaluation Metrics of clusters.....	37
Figure 5.2.3.1 Graph of Evaluation Metrics with variations of K (19 august 2016).....	38
Figure 5.2.3.2 Graph of Evaluation Metrics with variations of I (19 august 2016).....	39
Figure 5.2.3.3 Graph of Evaluation Metrics with variations of K (August 26, 2016).....	40
Figure 5.2.3.4 Graph of Evaluation Metrics with variations of I (August 26, 2016).....	40
Figure 5.2.3.5 Graph of Completion Time of Clustering Process with different values of I.....	41
Figure 5.2.3.6 Graph of Completion Time of Clustering Process with different values of K.....	42

## **List of Tables**

Table 5.2.1 List of Data Sets.....	31
Table 5.2.3.1 Evaluation Table for I=12, and Variations of K .....	38
Table 5.2.3.2 Evaluation Table for K=12, and Variations of I.....	38
Table 5.2.3.3 Evaluation Table for I=12, and Variations of K .....	39
Table 5.2.3.4 Evaluation Table for K=12, and Variations of I.....	39
Table 5.2.3.5 Completion Time of Clustering Process with different values of I.....	41
Table 5.2.3.6 Completion Time of Clustering Process with different values of K.....	41

## List of Abbreviations

<b>Abbreviations</b>	<b>Full Form</b>
IR	Information Retrieval
IE	Information Extraction
HTML	Hypertext Markup Language
Tf-idf	Term frequency-Inverse Document Frequency
MVC	Model -View -Controller
CSS	Cascading Style Sheets
SQL	Structural Query Language
HAC	Hierarchical Agglomerative Clustering

# CHAPTER 1

## INTRODUCTION

Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases [1, 2].

News Clustering System based on Text Mining is one of the implementation of Data Mining in which the similar type articles of different Newspapers are grouped together. The algorithms implemented for this work are as follows:

- News Extraction
- Document Parsing
- Document Preprocessing
- Document Representation
- Cosine Similarity Algorithm
- Document Clustering

The efficiency of K-means clustering algorithm for different values of K (initial number of seeds) has been analyzed to group the similar news. The relationship between the size of data (n) and the initial number of seeds (K) has been analyzed.

This this work entitled “**News Clustering System based on Text Mining**” is based on the approach of extracting information from the online news portals, i.e. information extraction (IE) and arranging them into clusters based on the similarity of the extracted information, i.e. clustering [3, 4]. The IE process and the clustering technique are the main focus points of this work.

Text mining is an important technique because it enables efficient analysis of existing knowledge. As explained by the authors in [5, 6], some of the advantages of implementing text mining include:

- Efficiency in terms of time.
- Unlocking hidden information and developing new knowledge.
- Exploring new horizons (research areas).
- Improved research and evidence base.
- Improving the research process and quality.

## **1.1 THESIS ORGANIZATION**

Introduction Part of this dissertation work focuses on the IR and the Data Mining along with the main processes of this work.

The rest of the material in this study is organized into five subsequent chapters.

Chapter 2 provides the background study required for this work. In this chapter, problem of lack of news clustering system is given, problem statement is formulated and main objective is mentioned.

Chapter 3 contains the previous work related to this dissertation in detail under literature review and research question is formulated. Proposed framework is described in detail in this chapter.

Chapter 4 provides the implementation of News Clustering System using Ruby on Rails.

Chapter 5 includes the collected data of news, and the performance measure of the system with different values of initial clusters seeds with table as well as graph.

At last, the concluding remarks and further enhancements are outlined in chapter 6.

## CHAPTER 2

### BACKGROUND STUDY AND PROBLEM FORMULATION

#### 2.1. BACKGROUND

##### 2.1.1. Information Extraction

Information Extraction has the goal of retrieving and storing structured data from natural language texts in order to improve corporate knowledge based (KB) processes. It is the type of information retrieval whose main theme is to automatically extract structured information from unstructured or semi-structured machine readable documents. IR retrieves relevant documents from collections, while IE extracts relevant information from documents [7, 8]. Hence the two techniques are complementary, and used in combination they can provide powerful tools for text processing. The tasks that IE system can perform are as follows:

- **Term analysis** – Identifies the terms in a document (For e.g. scientific research papers).
- **Named-Entity Recognition** – Identifies the names in a document (For e.g. names of people/organization).
- **Fact Extraction** – Identifies and extracts complex facts from documents.

##### 2.1.2 Web Crawler

A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Other terms for Web crawlers are ants, automatic indexers, bots, Web spiders, Web robots, or—especially in the FOAF community—*Web* scutters. This process is called Web crawling or spidering. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for sending spam).

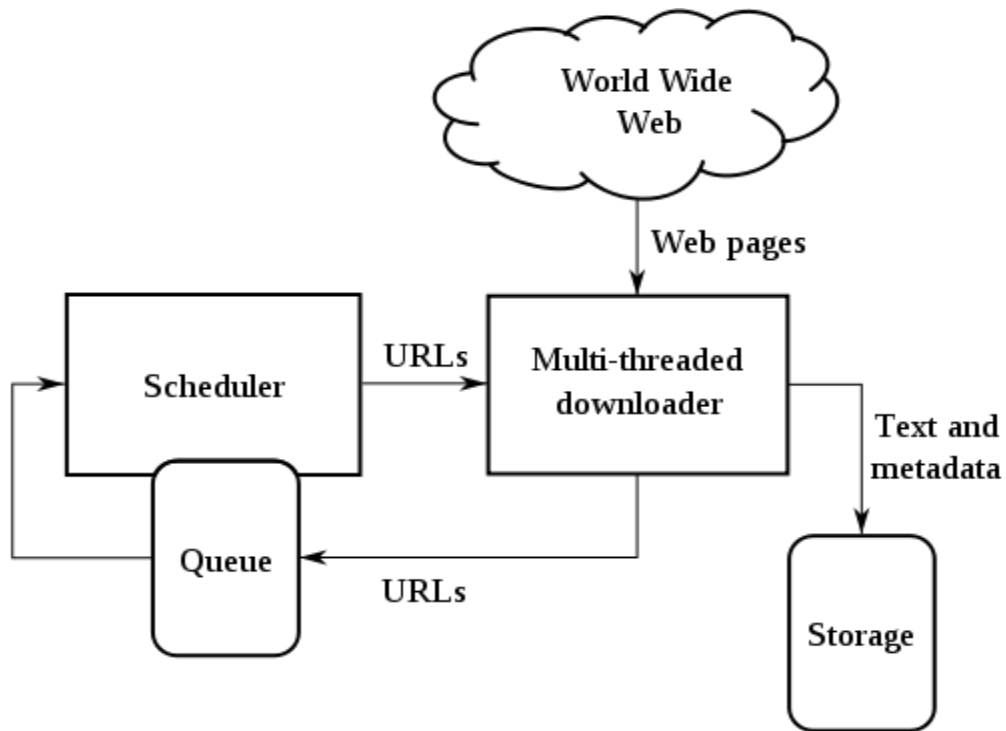


Figure 2.1.2.3 Architecture of Web Crawler [3]

### Web Crawler Architecture

The basic architecture of a web crawler is shown in the figure 2.1.2.2 given below. The crawler consists of several modules namely URL frontier, DNS resolution module, fetch module, parsing module, duplicate elimination module, URL filter and document finger print module.



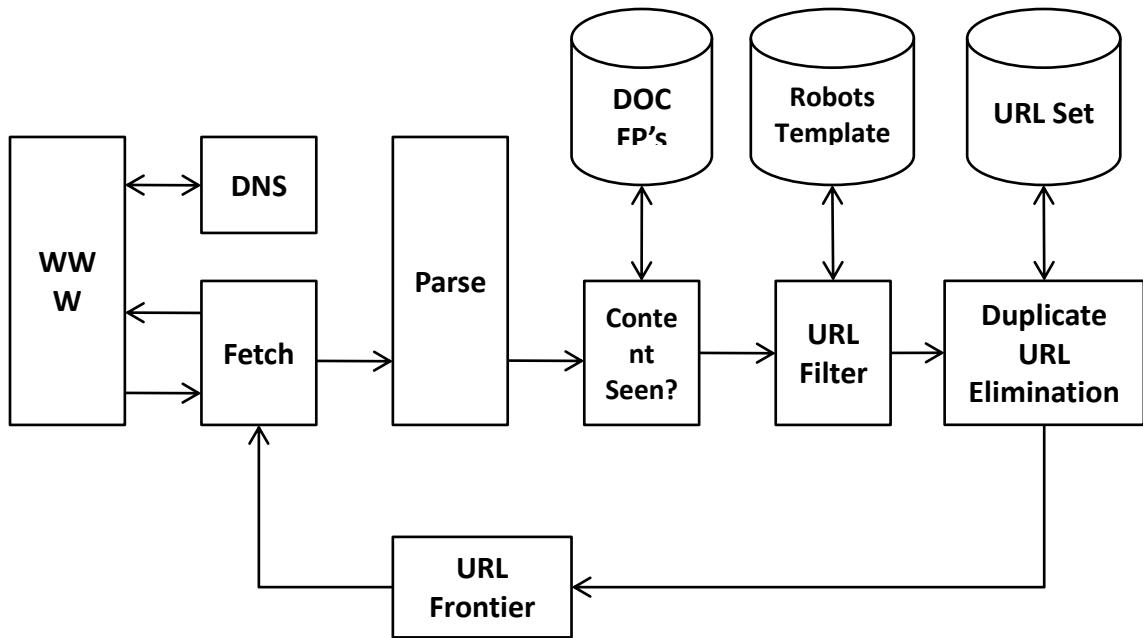


Figure 4.1.2.2 Basic Architecture of Web Crawler [3]

Each module has its own specific function or task to carry out, which are as follows:

- **URL Frontier** – It contains the URLs yet to be fetched.
- **Fetch Module** – It retrieves the web pages at the URL.
- **DNS Resolution Module** – It determines the web server from which to fetch the page specified by the URL.
- **Parsing Module** – Extracts the text and set of links from a fetched web page.
- **Duplicate Elimination Module** – It determines whether an extracted link is already in the URL frontier.
- **URL Filter** – It determines whether the extracted link should be excluded from the URL frontier.
- **Document Finger Print Module** – It checks whether a web page with the same content has been already seen at another URL.

### The Crawling Operation

The basic operations of a crawler are as follows:

- The crawler begins with one or more URLs that constitute a seed set.
- It picks a URL from this seed set, and then fetches the web page at that URL.

- The extracted text is then fed to a text indexer.
- The extracted links (URLs) are then added to a URL frontier which at all-time consists of URLs whose corresponding pages have yet to be fetched by the crawler.
- Initially, the URL frontier contains the seed set, as pages are fetched, the corresponding URLs are deleted from the URL frontier.

### 2.1.3 Document Preprocessing

#### a. Tokenization

Before any further processing can be done on a document, its text must be segmented into words and sentences; such task is known as tokenization. Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces called tokens. In other words, tokenization is the process of breaking up a stream of text into words, phrases, symbols or other meaningful elements called tokens [8, 9]. A token is an instance of characters in some particular document that are grouped together as a useful semantic unit for processing. Hence, these tokens become input for further processing such as parsing, text mining, etc. During this phase of document preprocessing, all the remaining text is parsed and lowercased. Also, all the punctuations are removed.

For example: Input : Please, Lend me your ears.

Output : |Please| |Lend| |me| |your| |ears|

In the above example, the sentence “*Please, Lend me your ears.*” is chopped up into 5 tokens i.e. *Please, Lend, me, your, ears* excluding the punctuation.

A tokenizer depends on simple heuristics such as the follows [10]:

- All nearby strings of alphabetic characters are part of one token. The same applies for numbers.
- Tokens are separated by whitespace characters, such as a space or line break or by punctuation characters.
- Punctuation and whitespace may or may not be included in the resulting list of tokens.

## **b. Stop Word Removal**

Every now and then, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary. Such words are called stop words. The process of excluding such words from documents is called stop word removal. The stop words, i.e. the occurrence of some very frequent words that does not carry information are removed since it reduces the quality of data mining.

The common strategy for determining a stop list is to sort the terms by collecting frequency and then to take out the most frequent terms and are then discarded during indexing. Using a stop list significantly reduces the number of postings that a system has to store. For example: a, am, are, and, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with, etc.

## **c. Porter Stemming Algorithm**

Stemming is the process of removing suffixes by automatic means is an operation which is especially useful in the field of information retrieval. In a typical IR environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abstract. Ignoring the issue of precisely where the words originate, we can say that a document is represented by a vector of words, or terms. Terms with a common stem will usually have similar meanings, for example:

CONNECT  
CONNECTED  
CONNECTING  
CONNECTION  
CONNECTIONS

Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

## 2.1.4 Clustering

The process of grouping a set of objects into classes, subsets or clusters of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The idea of clustering is to partition a free set of objects into clusters. Among many mechanisms of text mining, clustering is one of the most important techniques. There are number of clustering algorithms which are used in different areas or fields. If clustering algorithms are to be used, two conditions are necessary; (i) an object representation and (ii) a similarity (or distance) measure between objects [8]. Clustering algorithm mainly focuses on distance based cluster analysis.

In context to this work, since text and documents are to be clustered i.e. text clustering, texts are considered as the objects. The goal of text clustering is to create a group of similar documents or document fragments like news items [5].

### 2.1.4.1 K-Means Clustering

The **K-means** clustering algorithm is known to be efficient in clustering large data sets. This clustering algorithm was developed by MacQueen , and is one of the simplest and the best known unsupervised learning algorithms that solve the well-known clustering problem. The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into K clusters, where K is a predefined or user-defined constant. The main idea is to define K centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended Jaccard) to all objects in that cluster.

#### Basic K-Means Algorithm

1. Choose K number of clusters to be determined
2. Choose K objects randomly as the initial cluster center
3. Repeat

- 3.1. Assign each object to their closest cluster centre.
- 3.2. Compute new cluster centres, i.e. Calculate mean points.
4. Until
  - 4.1. No changes on cluster centers (i.e. Centroids do not change location any more) OR
  - 4.2. No object changes its cluster (We may define stopping criteria as well)

### **Termination Condition**

We can apply one of the following termination conditions.

- A fixed number of iterations . This condition limits the runtime of the clustering algorithm, but in some cases the quality of the clustering will be poor because of an insufficient number of iterations.
- Assignment of documents to clusters does not change between iterations but runtimes may be unacceptably long.
- Centroids do not change between iterations.

### **Bad choice of initial seed**

In K-Means algorithm there is unfortunately no guarantee that a *global minimum* in the objective function will be reached, this is a particular problem if a document set contains many *outliers* , documents that are far from any other documents and therefore do not fit well into any cluster. Frequently, if an outlier is chosen as an initial seed, then no other vector is assigned to it during subsequent iterations. Thus, we end up with a *singleton cluster* (a cluster with only one document).

Effective heuristics for seed selection include:

1. Excluding outliers from the seed set.
2. Trying out multiple starting points.

#### **2.1.4.1 Cosine Similarity Algorithm**

**Cosine similarity** is a measure of similarity between two vectors by measuring the cosine of the angle between them. The cosine of 0 is 1, and less than 1 for any other angle; the lowest value of

the cosine is -1. The cosine of the angle between two vectors thus determines whether two vectors are pointing in roughly the same direction.

In data mining we can use this technique to find the similarity of the documents.

For example

d1 = i have to go to school.

d2 = i have to go to toilet.

The words of the first sentence are i , have, to, go , school and all the words frequency is 1 except to the words of the second sentence. The words of the second sentence are i, have, to, go, to, toilet and again all the words frequency is 1 and if we think n-dimensional space the points of the words in space is

1 [i , have , to , go , school , toilet] = [1,1,2,1,1,0]

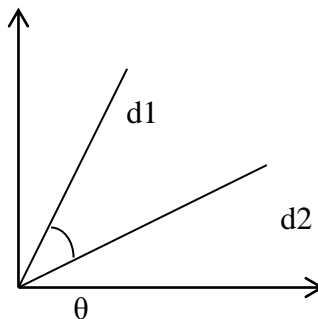
2 [i , have , to , go , school , toilet] = [1,1,2,1,0,1]

$$\cos \theta = \frac{1*1 + 1*1 + 2*2 + 1*1 + 1*0 + 0*1}{\sqrt{(1^2 + 1^2 + 2^2 + 1^2 + 1^2 + 0^2) + 1^2 + 1^2 + 2^2 + 1^2 + 0^2 + 1^2}}$$

In general

$$\text{Similarity}(d1,d2) = \cos(\theta) = \frac{\vec{d1} \cdot \vec{d2}}{|\vec{d1}| \cdot |\vec{d2}|}$$

So on the above two documents we will have altogether 6 dimensional vector space so the documents is represented as below



**Figure 2.1.4.1.1 Representation of document in Vector Space**

### 2.1.4.2 TF-IDF

Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

Tf-idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

#### **Example:**

Consider a document containing 100 words wherein the word *cat* appears 3 times. The term frequency (i.e., tf) for *cat* is then  $(3 / 100) = 0.03$ . Now, assume we have 10 million documents and the word *cat* appears in one thousands of these. Then, the inverse document frequency (i.e., idf) is calculated as  $\log(10,000,000 / 1,000) = 4$ . Thus, the Tf-idf weight is the product of these quantities:  $0.03 * 4 = 0.12$ .

## 2.2 PROBLEM FORMULATION

### 2.2.1 Problem Statement

Most of the users consider online news as resourceful web facility instead of reading the news by purchasing papers. Now a days there are numbers of websites available which provides daily national and international news. If a user wants to get more information on a concerned news article from a different source; for this purpose, the user will have to list out the sites containing articles on similar topic which can be tedious and time consuming. To overcome these problems, the concept of article mining is used to cluster the news on the WWW.

The information extraction process from various online news portals is the most challenging task in article mining. The main issue in article mining is to use short description of stories or

headings available. Such tasks are solved using simple strategies like analyzing the HTML code of the news's web-page and recognizing its pattern of display. News mining/Article mining mainly focuses to disambiguate information and to provide users with greater search experiences. This approach increases the quality of the results because [7]; the news items are short and contain relevant and descriptive key words. In general, article mining is based on a repository of web newspapers pages and extracting the items for these pages. The web pages are dynamic and changes continuously, hence to capture the information we retrieve the pages of the newspaper at regular intervals and store it in a database.

In this work, the news headlines from different sources of online web newspapers have been used. The keywords for each piece of news are extracted from the headlines which are further used to cluster similar news from a news data bank.

User have desire to read article in the newspaper that haven't been read already in another newspaper. But there is no grouping of similar type article from different newspapers. They have to visit all the news sites and have to search into all the respective newspapers' sites which is not reliable for them.

### **2.2.2 Objectives**

The main purposes of this dissertation are:

- To group the similar articles of different newspapers.
- To analyze the efficiency of K-means clustering algorithm to group the similar news with different values of K, and to analyze the relationship between number of documents N with number of iterations (I) and K.
- To reduce the complexity of searching similar news by visiting each website.

## **2.3 MOTIVATION**

News is today's most common sources for learning about current events. In addition, news may deal with topics of more long-term interest. It reflects and form societies', groups' and individuals' views of the world, fast or even instantaneous with the events triggering the reporting circumscribe. News is generally authored by people with journalistic training who



abide by journalistic standards regarding the style and language of reporting. Topics and ways of reporting are bed by general societal consensus and the policies of the news provider. The content of news basically involves text, pictures and additional content in other formats. The news items in an online news portal are a good source for studying the text mining techniques.

## **CHAPTER 3**

### **LITERATURE REVIEW AND METHODOLOGY**

#### **3.1 RELATED WORKS**

There are many approaches to text mining. Several researchers have implemented the concept of text mining.

This thesis includes three contributions: a survey of known clustering methods, an evaluation of human versus human results when grouping news articles in an event-centric manner, and last an evaluation of an incremental clustering algorithm [1,2]. In this work an information system has been proposed that will extract the main topics in the news archive in a weekly basis. By getting a weekly report, user can know what were the main news events in the past week [3].

In this paper work different clustering methods and their effectiveness has been compared for text document datasets for sentiment analysis. It results that K-means algorithm gives overall best results when used with Cosine Similarity considering all the factors that affects performance of Document Similarity Algorithm and Document Clustering Algorithm [4, 5]. This paper discusses the terms document and similarity in the given context and Apache Lucene tool which provides a foundation to build an information retrieval system for documents [6, 9]. This paper compares and analyzes the effectiveness of these measures in partial clustering for text document datasets. Experiments utilize the standard K-means algorithm and results on seven text document datasets and five similarity measures (Euclidian Distance, Cosine Similarity, Jaccard Coefficient, Pearson Correlation Coefficient, Averaged KullBack-Leibler Divergence) that have been most commonly used in text clustering [7]. This a real time news extraction system capable of identifying up-to-date “hot news” from large amounts of news reports on the internet [8].

In this HAC (Hierarchical Agglomerative Clustering) algorithm and Correlation similarity is used for any type of text document to display the most relevant document of the clusters [10]. This thesis is mainly focused on the use of text mining techniques and the K-means algorithm to create the clusters of similar news articles headlines. It is based on the text mining with primary focus on data mining and information extraction [12].

In order to improve on complete-page mining, it presents an approach based on extracting the individual news items from the web pages and mining these separately [13, 18]. This paper presents UPD Digital Library Miner, a software application for mining document collections of a digital library for topical structure discovery and topic-based similarities search between collection pairs, using topic modeling algorithm and Kullback-Leibler divergence measure [14]. This proposal work is made to improve the pruning of feature selection algorithm by clustering with distance boundaries and partitioning of uncertain probability distribution values [15]. This presents the comparison of two main document clustering techniques: agglomerative hierarchical clustering and K-means. For K-means a “standard” K-means algorithm and variant of K-means, “bisecting” K-means is used [16, 20]. In this paper, the similarity of two documents is gauged by using two string-based measures which are character-based method, n-gram is utilized to find fingerprint and Dice coefficient is used to match two fingerprints [17]. This survey discusses the existing works on text similarity through partitioning them into three approaches; String-based, Corpus-based and Knowledge-based similarities. Furthermore, samples of combination between these similarities are presented [19].

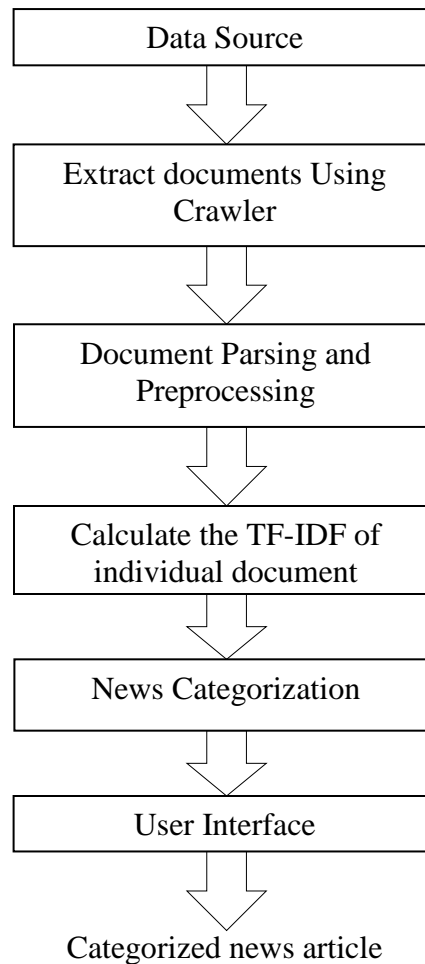
### **3.2 RESEARCH QUESTION**

News published in news portal vary from each other and people who read news desire to go through each of them in order have better and more clear information in that particular news headline. Therefore, it becomes a tiresome job for a person who has such a desire. The main objective of this work is to answer question of “how it is possible to view and compare same news which are published in different news portal into one single roof with the use of text mining”.

The overall purpose of this work can be summarized as following research questions:

- How text mining and clustering techniques can be used to generate required system?
- How it is possible to cluster similar news published in different portal into one single roof?
- Is there any relationship between the initial number of clusters (K) and the number of data i.e. number of news?

### 3.3 PROPOSED FRAMEWORK



**Figure 3.3.1. Proposed Framework for News Clustering**

#### 3.3.1 News Content Extraction

A web crawler extracts the news contents with their links and news heading from the respective news sites.

##### **Web Crawler:**

A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Other terms for Web crawlers are ants, automatic indexers, bots, Web spiders, Web robots, or—especially in the FOAF community—*Web* scutters.

Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches [5].

### **3.3.2 Parsing**

The HTML parser parses the unnecessary tags and links present in the extracted news contents. It also identifies the headings of the news [5, 6].

### **3.3.3 Document Preprocessing**

When the news contents with their links and news heading is extracted by crawler, it is further Preprocessed for text mining using document preprocessing techniques, which includes the following steps:

#### **Tokenization**

Tokenization is the process of chopping up a given stream of text or character sequence into words, phrases, symbols, or meaningful elements called tokens, which are input for the further processing of text mining [5,7]. The headings of different newspapers are tokenized. This process of tokenization is accomplished by using space to split the sequence of tokens.

#### **Stop Word Removal**

Those common words which would appear to be a little value in the documents matching, need to be excluded, are stop words and the process of excluding such words is called stop word removal [5, 7, 8]. The stop word such as a, an, the, and prepositions is created and the tokens contained in the stop word list are discarded.

#### **Lemmatization**

Lemmatization is the process of reducing the derived forms of a word to its base form called lemma so that they can be analyzed as a single term [5, 7, 8]. Stemming is a preferred method for lemmatization.

#### **Porter Stemming Algorithm:**

Stemming is the process of removing suffixes which is especially useful in the field of

information retrieval. In a typical IR environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abstract. Ignoring the issue of precisely where the words originate, we can say that a document is represented by a vector of words, or terms. The suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous [5, 7, 8].

### 3.3.4 Document Representation

Document representation is a key process in the information retrieval systems. To extract the relevant documents from the large collection of documents, it is very important to transfer the documents to vector form. The vector space model is popular algebraic model to represent the textual document as vectors. Using the vector space model documents are represented with the term frequency, inverse document frequency (tf-idf) weighting scheme [5, 14,15].

Example: The one of the news heading of August 26, 2016 is **“President, PM condolence to death in Chitwan bus fall”**

tokens: | president | PM | condolence | to | death | in | chitwan | bus | fall |

After stop word removal and stemming: | president | pm | condolence | death | chitwan | bus | fall |

Vector: [0.01555, 0.02358, 0.02137, 0.02028, 0.01867, 0.01806, 0.03129]

### 3.3.5 Document Clustering

After the construction of document vector, the clustering process is carried out using K-means clustering algorithm. The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into K clusters, where K is a predefined or user-defined constant [20]. The main idea is to define K centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended Jaccard) to all objects in that cluster [9, 10, 11, 19]. Here, cosine measure is used to compute which document centroid is closest to a given document.

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 TOOLS USED

All the algorithms have been implemented using ruby language in ruby on rails framework with the partial use of ruby libraries.

##### 4.1.1 Resource Requirements

The resources used to complete this work are as follows:

- My-SQL for database design.
- Web Server for testing proposed system.
- Platform: Windows XP or greater.
- Internet Explorer, Google Chrome, Mozilla Firefox or any other browser to view results.
- Programming Languages used: Ruby.
- Interface Design: HTML, CSS.

##### 4.1.2 Programming Language

###### Ruby

Ruby programming language has been used for the implementation of this work. **Ruby** is a [dynamic, reflective, object-oriented, general-purpose programming language](#). Everything is an [expression](#) (even [statements](#)) and everything is executed [imperatively](#) (even [declarations](#)) in ruby. It has an elegant syntax that is natural to read and easy to write.

Ruby 2.3.1 has been used in ruby on rails web framework to complete this work. **Ruby on Rails**, or simply **Rails**, is a server-side [web application framework](#) written in [Ruby](#). Rails is a [model–view–controller](#) (MVC) framework, providing default structures for a [database](#), a [web service](#), and [web pages](#). It encourages and facilitates the use of [web standards](#) such as [HTML](#), [CSS](#) and [JavaScript](#) for display and user interfacing.



## MySQL

MySQL is an [open-source relational database management system](#) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](#)' daughter, and "[SQL](#)", the abbreviation for [Structured Query Language](#). MySQL is also used in many high-profile, large-scale [websites](#), including [Google](#). MySQL is written in [C](#) and [C++](#).

## CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. It's most common application is to style web pages written in HTML and XHTML. CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed.

## HTML

HTML, which stands for Hypertext Markup Language, is the predominant markup language for web pages. HTML uses markup tags to describe web pages. HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like `<html>`), within the web page content. HTML tags normally come in pairs like `<h1>` and `</h1>`. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). The purpose of a web browser is to read HTML documents and compose them into visual web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior or HTML markup.

## 4.2 DATA SOURCE MODULE

```
SOURCES = {  
  "The Himalayan Times" => "http://www.thehimalayantimes.com",  
  "Republica" => "http://www.myrepublica.com",  
  "The Kathmandu Post" => "http://kathmandupost.ekantipur.com",  
  "Nepal News" => "http://www.nepalnews.com",  
  "The Rising Nepal" => "http://therisingnepal.org.np/",  
  "Online Khabar" => "http://english.onlinekhabar.com/",  
  "NP News Portal" => "http://www.npnewsportal.com/",  
  "Nepali Headlines" => "http://nepaliheadlines.com/"  
}
```

```
SOURCES.each do |name, link|  
  puts "Creating news source: #{name}"  
  NewsSource.create(name: name, link: link)  
end
```

## 4.3 News Extraction Module

```
require 'open-uri'  
  
namespace :crawler do  
  
  desc 'Gets news URLs'  
  
  task run: :environment do  
  
    NewsSource.all.each do |source|  
  
      doc = Nokogiri::HTML(open(source.link))  
  
      links = doc.css('a').select { |l| l.content.length > 20 }  
  
      puts "News Source: #{source.name}"  
  
      puts "Total Headings: #{links.count}"  
  
      links.each do |link|
```

```

    href = link.attributes['href'].value rescue next
    permalink = href =~ URI::regexp ? href : source.link + href
    source.news.find_or_create_by(title: link.content, link: permalink)
  end
end
end

desc "Updates lemma"
task lemmatize: :environment do
  News.all.each do |news|
    news.lemma = news.lemmatize
    news.save
  end
end

desc "Save generated tokens based on News items"
task tokenize: :environment do
  Token.generate
end
end

```

#### **4.4 DOCUMENT PREPROCESSING MODULE**

```

class Token < ActiveRecord::Base
  belongs_to :news_source
  belongs_to :news
  def self.generate
    News.all.each do |news|

```

```

news.lemma.each do |lema|

  token = Token.find_by(word: lema.downcase, news_source: news.news_source, news:
news)

  if token

    token.increment!(:frequency)

  else

    Token.create(word: lema.downcase, news_source: news.news_source, frequency: 1, news:
news)

  end

end

end

end

end

def self.distinct

  @@distinct ||= Token.pluck(:word).uniq

end

def tf_idf

  tf*idf

end

end

```

```

class News < ActiveRecord::Base

  serialize :lemma, Array

  STOP_WORDS = [
"a","about","above","after","again","against","all","am","an","and","any","are","arn't","as","at","
be","because","been","before","being","below","between","both","but","by","can","can't","cann
ot","could","couldn't","did","didn't","do","does","doesn't","doing","don't","down","during","eac
h","few","for","from","further","had","hadn't","has","hasn't","have","haven't","having","he","he'
d","he'll","he's","her","here","here's","hers","herself","him","himself","his","how","how's","i","i'

```

```
d", "i'll", "i'm", "i've", "if", "in", "into", "is", "isn't", "it", "it's", "its", "itself", "let's", "me", "more", "most", "mustn't", "my", "myself", "no", "nor", "not", "of", "off", "on", "once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shan't", "she", "she'd", "she'll", "she's", "uld", "shouldn't", "so", "some", "such", "than", "that", "that's", "the", "their", "theirs", "them", "themselves", "then", "there", "there's", "these", "they", "they'd", "they'll", "they're", "they've", "this", "those", "through", "to", "too", "under", "until", "up", "very", "was", "wasn't", "we", "we'd", "we'll", "we're", "we've", "were", "weren't", "what", "what's", "when", "when's", "where", "where's", "which", "while", "who", "who's", "whom", "why", "why's", "with", "won't", "would", "wouldn't", "you", "you'd", "you'll", "you're", "you've", "your", "yours", "yourself", "yourselves"]
```

```
belongs_to :news_source
```

```
belongs_to :cluster
```

```
has_many :tokens
```

```
def self.lemmatizer
```

```
  @@lemmatizer ||= Lemmatizer.new
```

```
end
```

```
def tokenize
```

```
  title.gsub(/[\^w\s]/, "").to_s.split(" ")
```

```
end
```

```
def without_stop_words
```

```
  tokenize - News::STOP_WORDS
```

```
end
```

```
def stem
```

```
  without_stop_words.map(&:stem)
```

```
end
```

```
def lemmatize
```

```
  stem.map { |s| News.lemmatizer.lemma(s) }
```

```
end
```

```
namespace :token do
```

```
  namespace :tf do
```

```
    desc "Calculates TF"
```

```
    task calculate: :environment do
```

```

Token.all.each do |token|
  news_count = token.news_source.news.count
  token.tf = token.frequency.to_f / news_count.to_f
  token.save
end
end
end
namespace :idf do
  desc "Calculates IDF"
  task calculate: :environment do
    Token.all.each do |token|
      news_count = News.count
      global_token_frequency = Token.where(word: token.word).map(&:frequency).sum
      token.idf = Math.log10(news_count/global_token_frequency)
      token.save
    end
  end
end
end
end

```

## 4.5 CLUSTERING MODULE

```

def vector
  news_tokens = tokens.select { |t| t.news_source_id == news_source_id &&
lemma.map(&:downcase).include?(t.word) }
  Token.distinct.map do |t|
    token = news_tokens.select{ |l| l.word == t }.first
    token ? token.tf_idf : 0.0
  end
end
end

```

```

def similarity(news1)
  vector_1 = self.vector
  vector_2 = news1.vector
  product = vector_1.zip(vector_2).map {|p| p.map(&:to_f).inject(:*)}.compact.sum
  length_1 = Math.sqrt(vector_1.map{|i| i ** 2}.sum)
  length_2 = Math.sqrt(vector_2.map{|i| i ** 2}.sum)
  sim = product.to_f/(length_1*length_2).to_f
  sim.nan? ? 0.0 : sim
end
end
namespace :clusterify do
  desc "Clusterify"
  task run: :environment do
    # Random centers
    k = ENV['K'] || 12
    k = k.to_i
    i = ENV['I'] || 7
    i = i.to_i
    Cluster.delete_all
    News.update_all(cluster_id: nil)
    puts "Creating initial centers"
    initial_centers = News.all.sample(k).to_a
    k.times do |i|
      initial_center = initial_centers[i]
      cluster = Cluster.create(name: "Cluster_#{i+1}")
      initial_center.cluster = cluster
      initial_center.save
    end
    puts "Clustering items to the initial centers"
    clusters = Cluster.all.to_a
    news = News.all.includes(:tokens).to_a

```

```

(news - initial_centers).each do |n|
  rank = { }
  initial_centers.each do |initial_center|
    rank[initial_center.cluster_id] = n.similarity(initial_center)
  end
  highest_matching_key = rank.key(rank.values.max)
  unless rank[highest_matching_key].zero?
    n.cluster_id = highest_matching_key
  else
    n.cluster_id = clusters.sample.id
  end
  n.save
end
# Calculate new centers
i.times do |i|
  puts "Start of Iteration #{i}"
  clusters.each do |cluster|
    cluster_news = cluster.news.reload.includes(:tokens)
    cluster_news_count = cluster_news.count
    cluster.mean = cluster_news.map(&:vector).transpose.map(&:sum).collect { |n|
n/cluster_news_count }
    puts "Cluster: #{cluster.name}"
    puts "Sum of mean: #{cluster.mean.sum}"
  end
  news.each do |n|
    rank = { }
    clusters.each do |c|
      rank[c.id] = n.similarity(c)
    end
    highest_matching_key = rank.key(rank.values.max)
    n.cluster_id = highest_matching_key
  end
end

```



```

    n.save
  end
  puts "End of Iteration #{i}"
end
end
end

```

## 4.6 EVALUATION MODULE

```

namespace :evaluation do
  desc "Calculating"
  task run: :environment do
    threshold = 0.12
    puts "Threshold: #{threshold}"
    Cluster.all.each do |cluster|
      puts "Calculating similarity between news of cluster: #{cluster.name}"
      cluster_news = cluster.news.reload.includes(:tokens)
      cluster_news_count = cluster_news.count
      cluster.mean = cluster_news.map(&:vector).transpose.map(&:sum).collect { |n|
n/cluster_news_count }
      similar_count = 0
      cluster.news.each do |news|
        #puts "News: #{news.title}"
        #puts "Similarity: #{news.similarity(cluster)}"
        similar_count += 1 if news.similarity(cluster) > threshold
      end
      news_count = cluster.news.count
      puts "Total news: #{news_count}"
      puts "Similar news: #{similar_count}"
      precision = similar_count.to_f/news_count.to_f
      recall = similar_count.to_f/(similar_count + 10.0).to_f
      cluster.update(precision: precision, recall: recall)
    end
  end
end

```

```

    end
  end
end
class Cluster < ActiveRecord::Base
  attr_accessor :mean
  has_many :news
  def agg_mean
    total_news = news.reload.includes(:tokens)
    total_news_count = total_news.count
    total_news.map(&:vector).transpose.map(&:sum).collect { |n| n/total_news_count }
  end
  def f_measure
    (2*recall.to_f*precision.to_f)/(precision.to_f + recall.to_f)
  end
  alias_method :vector, :mean
end

```

## CHAPTER 5

### DATA COLLECTION AND ANALYSIS

#### 5.1 DATA COLLECTION

In order to complete this work on article mining, data collection is a must. The data is collected or extracted from the HTML source code of various URL's related to the online Nepali news portal. Here, the news items or headlines are considered as the data used and are needed to be extracted. Besides the news item, other fragments such as general information about the newspaper, advertisements and links to regular columns like weather, horoscope, etc are not considered.

The extraction of news item (data) from various newspapers is done by the web crawler. The crawler is designed using simple strategies that are based on analyzing the HTML code of the web pages.

##### 5.1.1 Sources

The web consists of a number of websites which provides daily news. These website are called web newspapers or online news portals. These web newspapers are the source from which the data (i.e. headings) are extracted for this this work. There are 8 web newspaper websites from which the data was extracted.

The websites are:

- The Himalayan Times ([www.thehimalayantimes.com](http://www.thehimalayantimes.com))
- Republica ([www.myrepublica.com](http://www.myrepublica.com))
- The Kathmandu Post ([www.ekantipur.com/tkp/](http://www.ekantipur.com/tkp/))
- Nepal News (<http://www.nepalnews.com>)
- The Rising Nepal (<http://therisingnepal.org.np>)
- Online (<http://english.onlinekhabar.com>)
- NP News Portal (<http://www.npnewsportal.com>)
- Nepali Headlines (<http://nepaliheadlines.com>)

The web newspapers may contain topics that are not necessary or are unrelated such as banners, links, weather, images, etc. These set of components are discarded from being extracted. In this work, information extraction process has mainly two tasks: (a) crawl the online news portal to fetch the page of interest and (b) extract the news by parsing the HTML content.

### 5.1.2 News Data

All the news data are secondary data collected from the eight news portal listed above.

**Table 5.2.1 List of Data Sets**

S.N	Day	Source	News	Total News
		Himalayan Times	104	
		Republica	116	
		The Kathmandu Post	140	
1	19 August 2016	Nepal News	159	<b>638</b>
		The Rising Nepal	61	
		Online Khabar	58	
		Himalayan Times	103	
		Republica	115	
		The Kathmandu Post	140	
2	20 August 2016	Nepal News	157	<b>634</b>
		The Rising Nepal	62	
		Online Khabar	57	
		Himalayan Times	104	
		Republica	112	
		The Kathmandu Post	139	
3	21 August 2016	Nepal News	158	<b>639</b>
		The Rising Nepal	71	
		Online Khabar	55	
		Himalayan Times	103	
4	22 August 2016	Republica	118	<b>641</b>

		The Kathmandu Post	136	
		Nepal News	155	
		The Rising Nepal	71	
		Online Khabar	58	
		Himalayan Times	108	
		Republica	115	
		The Kathmandu Post	140	
		Nepal News	162	
5	23 August 2016	The Rising Nepal	62	<b>803</b>
		Online Khabar	58	
		NP News Portal	98	
		Nepali Headlines	60	
		Himalayan Times	103	
		Republica	116	
		The Kathmandu Post	141	
		Nepal News	154	
6	25 August 2016	The Rising Nepal	69	<b>798</b>
		Online Khabar	58	
		NP News Portal	97	
		Nepali Headlines	60	
		Himalayan Times	104	
		Republica	119	
		The Kathmandu Post	140	
		Nepal News	158	
7	26 August 2016	The Rising Nepal	62	<b>791</b>
		Online Khabar	48	
		NP News Portal	100	
		Nepali Headlines	60	

News Source: The Kathmandu Post  
Total Headings: 143  
Heading: 12th South Asian Games  
Heading: Click here to see suggestions from google  
Heading: Constitution amendment proposal unnecessarily linked with India: PM Dahal  
Heading: House committee directs govt not to dole medical expenses to ex-VIPs arbitrarily  
Heading: Also read: Now even ex-MPs want state privileges  
Heading: Nepal Women's Association convention kicks off  
Heading: NRB governer urges parents not to send their children to foreign countries for work  
Heading: UK, Nepal sign MoU for Nepal's health sector reform  
Heading: Nepal medical education bill presented in House  
Heading: 200 students get bicycles on children's day  
Heading: Consensus reached to set up 13 local units in Morang  
Heading: PM Dahal informs parliament about his India visit  
Heading: Ganga Maya breaks her fast  
Heading: Cases of human trafficking on the rise: Reports  
Heading: Committee formed to probe NAC Airbus incidents  
Heading: Let's stop putting out begging bowl: Bhattarai to Dahal  
Heading: Secy shake-ups stoke policy instability fears  
Heading: Three-member panel to pick names for NEA chief  
Heading: Committee formed to probe NAC Airbus incidents  
Heading: Committee formed to probe NAC Airbus incidents  
Heading: NAC jet stops safely after losing a wheel on landing  
Heading: Arun III land acquisition snags on compensation  
Heading: BIBEK SUBEDI in KATHMANDU & DIPENDRA SHAKYA in SANKHUWASABHA  
Heading: OPPO's F1s hits Nepali market  
Heading: Sellers unveil offers to lure customers  
Heading: Process to name NTGC chief executive begins  
Heading: Process to name NTGC chief executive begins  
Heading: UK, Nepal sign MoU for Nepal's health sector reform  
Heading: Consensus reached to set up 13 local units in Morang  
Heading: Quake victims spend housing Aid on clothes, electronics  
Heading: Herbs smuggling goes unchecked  
Heading: Chure forest encroached  
Heading: Landslide blocks traffic at Rasuwagadhi-Syaphru road  
Heading: Cases of human trafficking on the rise: Reports

Figure5.2.1.1 Screenshot of Sample News headings extracted from respective portal

"plant", "decis", "to", "launch", "protest", "draw", "flak",  
 "fdi", "commit", "plung", "83", "per", "cent", "gold", "edg",  
 "up", "on", "soft", "dollar", "brexit", "concern", "eas", "far",  
 "expert", "stress", "on", "social", "transform",  
 "entrepreneurship", "confer", "soon", "emirat", "sale", "offic",  
 "oil", "price", "fall", "first", "time", "three", "dai", "asia",  
 "share", "paus", "sterl", "stand", "tall", "as", "brexit",  
 "vote", "loom", "nrb", "win", "us", "legal", "battl",  
 "repossess", "1", "million", "india", "unveil", "broad",  
 "foreign", "invest", "reform", "after", "central", "bank",  
 "chief", "exit", "germani", "beat", "northern", "ireland", "to",  
 "advanc", "as", "group", "winner", "blaszczykowski", "goal",  
 "cement", "poland", "place", "last", "16", "shami", "pink",  
 "over", "condit", "daynight", "test", "us", "face", "argentina",  
 "messi", "copa", "semi", "not", "overwhelm", "rip", "shirt",  
 "burst", "ball", "as", "franc", "draw", "00", "with", "swiss",  
 "misfir", "ronaldo", "leav", "portug", "danger", "place",  
 "messi", "equal", "record", "as", "argentina", "chile",  
 "advanc", "blake", "live", "ryan", "reynold", "and", "i", "ar",  
 "offici", "breeder", "drake", "view", "spend", "seventh",  
 "straight", "week", "atop", "billboard", "chart", "nbc", "to",  
 "unveil", "new", "theme", "song", "sundai", "night", "footbal",  
 "leonardo", "dicaprio", "be", "order", "depos", "over", "wolf",  
 "of", "wall", "street", "led", "zeppelin", "page", "testifi",  
 "to", "stairwai", "and", "mari", "poppin", "song", "similar",  
 "riff", "rift", "over", "led", "zeppelin", "stairwai", "return",  
 "to", "court", "led", "zeppelin", "lawyer", "ask", "judg", "to",  
 "toss", "stairwai", "case", "fiat", "chrysler", "to",  
 "investig", "crash", "that", "kill", "star", "trek", "actor",  
 "us", "to", "help", "fund", "technolog", "elimin", "zika",

Figure 5.2.1.2 Screenshot of Sample terms of news before preprocessing

"sale", "agreement", "iran", "air", "ga", "plant", "decis",  
 "protest", "draw", "flak", "fdi", "commit", "plung", "83",  
 "per", "cent", "gold", "up", "soft", "dollar", "brexit",  
 "concern", "eas", "far", "expert", "stress", "social",  
 "transform", "entrepreneurship", "confer", "soon", "emirat",  
 "offic", "oil", "price", "fall", "first", "time", "asia",  
 "share", "paus", "sterl", "stand", "tall", "as", "loom", "nrb",  
 "us", "legal", "battl", "repossess", "1", "million", "india",  
 "unveil", "broad", "foreign", "invest", "reform", "exit",  
 "beat", "northern", "ireland", "advanc", "group", "winner",  
 "blaszczykowski", "goal", "cement", "poland", "place", "last",  
 "16", "shami", "pink", "condit", "daynight", "test", "face",  
 "argentina", "messi", "copa", "semi", "not", "overwhelm", "rip",  
 "shirt", "burst", "ball", "franc", "00", "swiss", "misfir",  
 "ronaldo", "leav", "portug", "danger", "equal", "record",  
 "chile", "blake", "live", "ryan", "reynold", "and", "i", "ar",  
 "breeder", "drake", "view", "spend", "seventh", "straight",  
 "week", "atop", "billboard", "chart", "nbc", "theme", "song",  
 "sundai", "night", "footbal", "leonardo", "dicaprio", "order",  
 "depos", "wolf", "wall", "street", "led", "zeppelin", "page",  
 "testifi", "stairwai", "mari", "poppin", "similar", "riff",  
 "rift", "return", "lawyer", "ask", "judg", "toss", "fiat",  
 "chrysler", "investig", "that", "star", "trek", "actor", "help",  
 "technolog", "elimin", "zika", "blood", "suppli", "scienc",  
 "sweet", "electric", "trick", "mai", "lead", "le", "fat",

Figure 5.2.1.3 Screenshot of Sample terms of news after preprocessing

## 5.2 EXPERIMENTAL RESULT

### 5.2.1 Experimental Setup

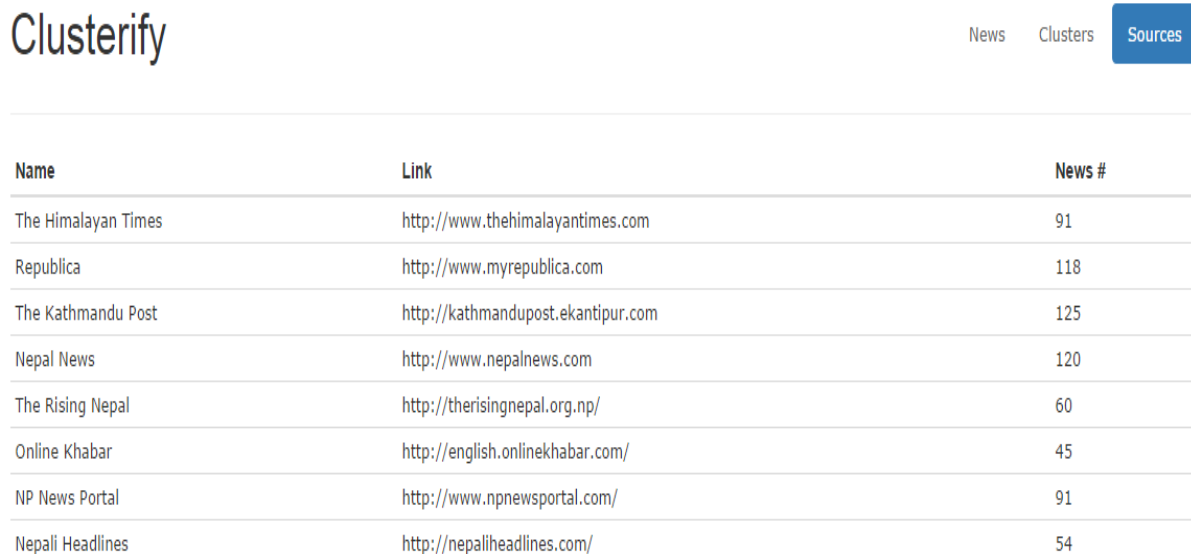
The aim is to experimentally verify the efficiency of proposed system of news clustering and to analyze the relationship of K with number of data.

The experiments were performed with Intel (R) Core (TM) i5 – M430 CPU @ 2.27 GHz 2.27 GHz of 4 GB RAM in 64-bit Windows 7 Operating System.

Clustering was conducted with Iteration I=12 and variations of K from 5 to 15, and with K=12, variations of I from 5 to 15 using the above datasets listed in **Table 5.2.1**. The efficiency was measured in terms of precision, recall and F-measure and running time.

### 5.2.2 Sample Output

Here are some snapshot of output of clustering testing at dated 26<sup>th</sup> August 2016 with K=12 and I=12.



The screenshot shows the Clusterify application interface. At the top left is the logo 'Clusterify'. On the top right, there are three buttons: 'News', 'Clusters', and 'Sources' (which is highlighted in blue). Below the buttons is a table with three columns: 'Name', 'Link', and 'News #'. The table lists eight news sources with their respective links and the number of news items.

Name	Link	News #
The Himalayan Times	<a href="http://www.thehimalayantimes.com">http://www.thehimalayantimes.com</a>	91
Republica	<a href="http://www.myrepublica.com">http://www.myrepublica.com</a>	118
The Kathmandu Post	<a href="http://kathmandupost.ekantipur.com">http://kathmandupost.ekantipur.com</a>	125
Nepal News	<a href="http://www.nepalnews.com">http://www.nepalnews.com</a>	120
The Rising Nepal	<a href="http://therisingnepal.org.np/">http://therisingnepal.org.np/</a>	60
Online Khabar	<a href="http://english.onlinekhabar.com/">http://english.onlinekhabar.com/</a>	45
NP News Portal	<a href="http://www.npnewsportal.com/">http://www.npnewsportal.com/</a>	91
Nepali Headlines	<a href="http://nepalihheadlines.com/">http://nepalihheadlines.com/</a>	54

Figure 5.2.2.1 Screenshot of Output of total News of each portal.



Source	Title	Link
The Himalayan Times	House meeting postponed owing to lack of quorum	<a href="http://thehimalayantimes.com/kathmandu/house-meet-postponed-owing-lack-quorum/">http://thehimalayantimes.com/kathmandu/house-meet-postponed-owing-lack-quorum/</a>
The Himalayan Times	Education Minister expresses discontent over establishment of new universities	<a href="http://thehimalayantimes.com/kathmandu/education-minister-expresses-discontent-establishment-new-universities/">http://thehimalayantimes.com/kathmandu/education-minister-expresses-discontent-establishment-new-universities/</a>
The Himalayan Times	Kamal Thapa demands secularism removed from Constitution	<a href="http://thehimalayantimes.com/nepal/kamal-thapa-demands-secularism-removed-constitution/">http://thehimalayantimes.com/nepal/kamal-thapa-demands-secularism-removed-constitution/</a>
The Himalayan Times	Jajarkot fire victim dies in course of treatment	<a href="http://thehimalayantimes.com/nepal/jajarkot-fire-victim-dies-course-treatment/">http://thehimalayantimes.com/nepal/jajarkot-fire-victim-dies-course-treatment/</a>
The Himalayan Times	Pedestrian killed in road mishap in Morang	<a href="http://thehimalayantimes.com/nepal/pedestrian-killed-road-mishap-morang/">http://thehimalayantimes.com/nepal/pedestrian-killed-road-mishap-morang/</a>
The Himalayan Times	Current coalition to stay intact till federal polls, says PM	<a href="http://thehimalayantimes.com/kathmandu/current-coalition-stay-intact-till-federal-polls-pushpa-kamal-dahal/">http://thehimalayantimes.com/kathmandu/current-coalition-stay-intact-till-federal-polls-pushpa-kamal-dahal/</a>
The Himalayan Times	21 killed as bus plunges off bridge in India	<a href="http://thehimalayantimes.com/world/21-killed-bus-plunges-off-bridge-india/">http://thehimalayantimes.com/world/21-killed-bus-plunges-off-bridge-india/</a>
The Himalayan Times	13 injured in Jhapa bus accident	<a href="http://thehimalayantimes.com/nepal/13-injured-jhapa-bus-accident/">http://thehimalayantimes.com/nepal/13-injured-jhapa-bus-accident/</a>
The Himalayan Times	Truck plunges into Trishuli River in Chitwan, goes missing	<a href="http://thehimalayantimes.com/nepal/truck-plunges-into-trishuli-river/">http://thehimalayantimes.com/nepal/truck-plunges-into-trishuli-river/</a>
The Himalayan Times	Bus hit injures three at Basundhara	<a href="http://thehimalayantimes.com/kathmandu/bus-hit-injures-three-in-basundhara/">http://thehimalayantimes.com/kathmandu/bus-hit-injures-three-in-basundhara/</a>
The Himalayan Times	FAA warns airline passengers not to use Samsung smartphone	<a href="http://thehimalayantimes.com/science-technology/faa-warns-airline-passengers-not-use-samsung-smartphone-galaxy-note-7/">http://thehimalayantimes.com/science-technology/faa-warns-airline-passengers-not-use-samsung-smartphone-galaxy-note-7/</a>
The Himalayan Times	NASA probe blasts off on quest to collect asteroid samples	<a href="http://thehimalayantimes.com/science-technology/nasa-probe-osiris-rex-blasts-off-quest-collect-asteroid-bennu-samples/">http://thehimalayantimes.com/science-technology/nasa-probe-osiris-rex-blasts-off-quest-collect-asteroid-bennu-samples/</a>

Figure 5.2.2.2 Screenshot of Output News before clustering

Source	Title	Link
The Kathmandu Post	I will address Dr KC's demands: Health Minister Thapa	<a href="http://kathmandupost.ekantipur.com/news/2016-08-26/i-will-address-dr-kcs-demands-health-minister-thapa.html">http://kathmandupost.ekantipur.com/news/2016-08-26/i-will-address-dr-kcs-demands-health-minister-thapa.html</a>
The Rising Nepal	Lama crowned 'Miss Grand Nepal-2016' title	<a href="http://therisingnepal.org.np/news/13934">http://therisingnepal.org.np/news/13934</a>
Nepal News	Follow @nepalnews_com	<a href="https://twitter.com/@nepalnews_com">https://twitter.com/@nepalnews_com</a>
The Rising Nepal	NC finalises names of ministers	<a href="http://therisingnepal.org.np/news/13970">http://therisingnepal.org.np/news/13970</a>
The Kathmandu Post	Baby among seven killed in Fishtail Air chopper crash	<a href="http://kathmandupost.ekantipur.com/news/2016-08-09/baby-among-seven-killed-in-fishtail-air-chopper-crash.html">http://kathmandupost.ekantipur.com/news/2016-08-09/baby-among-seven-killed-in-fishtail-air-chopper-crash.html</a>
The Kathmandu Post	Pokemon Go-playing driver kills woman in Japan	<a href="http://kathmandupost.ekantipur.com/news/2016-08-25/pokemon-go-playing-driver-kills-woman-in-japan.html">http://kathmandupost.ekantipur.com/news/2016-08-25/pokemon-go-playing-driver-kills-woman-in-japan.html</a>
NP News Portal	jiban thapa magar and joylin accident talk with mother purna maya magar	<a href="http://www.npnewsportal.com/jiban-thapa-magar-and-joylin-accident-talk-with-mother-purna-maya-magar/">http://www.npnewsportal.com/jiban-thapa-magar-and-joylin-accident-talk-with-mother-purna-maya-magar/</a>
Republica	Priority over community-based industry: Minister Joshi	<a href="http://www.myrepublica.com/news/4525">http://www.myrepublica.com/news/4525</a>
Nepal News	Ambien cr generic 12.5 mg	<a href="http://www.halfscale.com/?b=1484">http://www.halfscale.com/?b=1484</a>
Nepal News	UN chief to convene meeting on refugees given current refugee crisis in Europe	<a href="http://www.nepalnews.com/index.php/world-archive/45644-un-chief-to-convene-meeting-on-refugees-given-current-refugee-crisis-in-europe">http://www.nepalnews.com/index.php/world-archive/45644-un-chief-to-convene-meeting-on-refugees-given-current-refugee-crisis-in-europe</a>

Figure 5.2.2.3 Screenshot of Output News after Clustering

Name	News #	Precision	Recall	F-Measure
Cluster_1	60	0.5	0.75	0.6
Cluster_2	65	0.615384615384615	0.8	0.6956521739130432
Cluster_3	63	0.634920634920635	0.8	0.7079646017699115
Cluster_4	91	0.857142857142857	0.886363636363636	0.8715083798882679
Cluster_5	74	0.756756756756757	0.848484848484849	0.8000000000000003
Cluster_6	43	0.767441860465116	0.767441860465116	0.767441860465116
Cluster_7	45	0.622222222222222	0.736842105263158	0.6746987951807228
Cluster_8	53	0.716981132075472	0.791666666666667	0.7524752475247528
Cluster_9	37	0.756756756756757	0.736842105263158	0.7466666666666668
Cluster_10	48	0.625	0.75	0.6818181818181818
Cluster_11	42	0.666666666666667	0.736842105263158	0.7000000000000002
Cluster_12	83	0.686746987951807	0.850746268656716	0.7599999999999998

Figure 5.2.2.4 Screenshot of Output of Evaluation Metrics of clusters

### 5.2.3 Evaluation Metrics

Following metrics were used for the analysis of the news clustering:

For cluster  $i$  and class  $j$ :

$$\text{Precision}(i,j) : \frac{n_{ij}}{n_j} \quad \text{and} \quad \text{Recall}(i,j) : \frac{n_{ij}}{n_i}$$

Where,  $n_{ij}$  is the number of members of class  $i$  in cluster  $j$ ,  $n_j$  is the numbers of members of cluster  $j$ , and  $n_i$  is the number of members of class  $i$ .

The F measure of cluster  $j$  and class  $i$  is then given by

$$\mathbf{F}(i,j) : \frac{2 * \text{Recall}(i,j) * \text{Precision}(i,j)}{\text{Precision}(i,j) + \text{Recall}(i,j)}$$

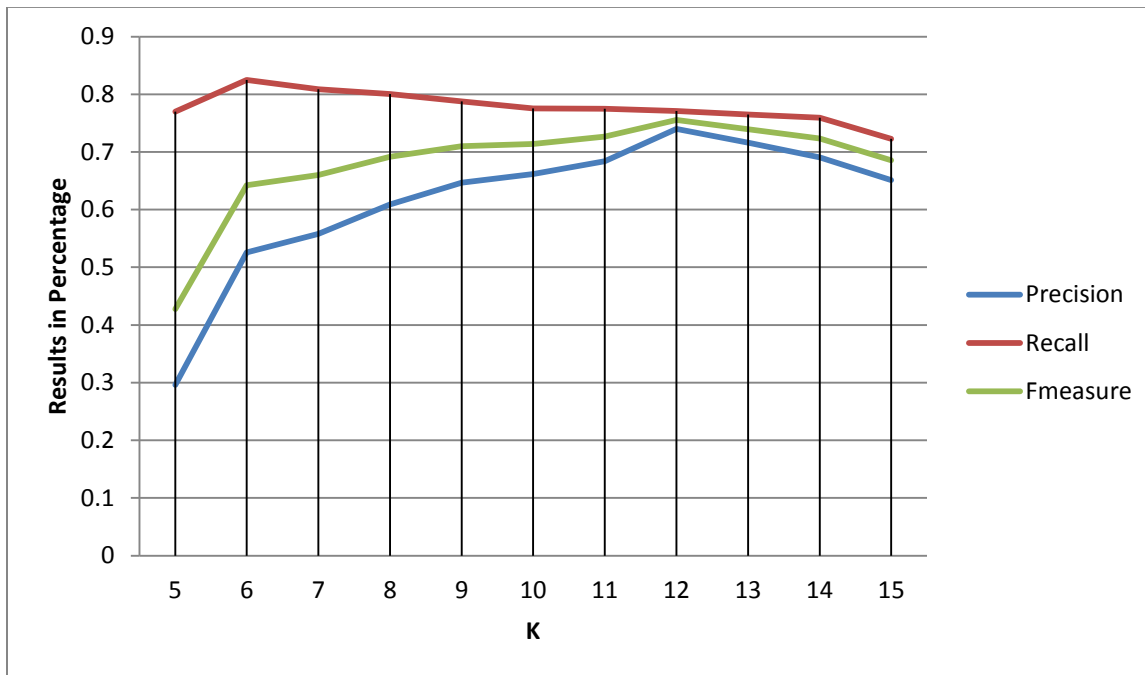
The sample results of the experiment (Day19 August 2016 with 638 News headings, Day 26 August 2016 with 791 News headings) have been shown as follows:

**Table 5.2.3.1 Evaluation Table for I=12,  
and Variations of K**

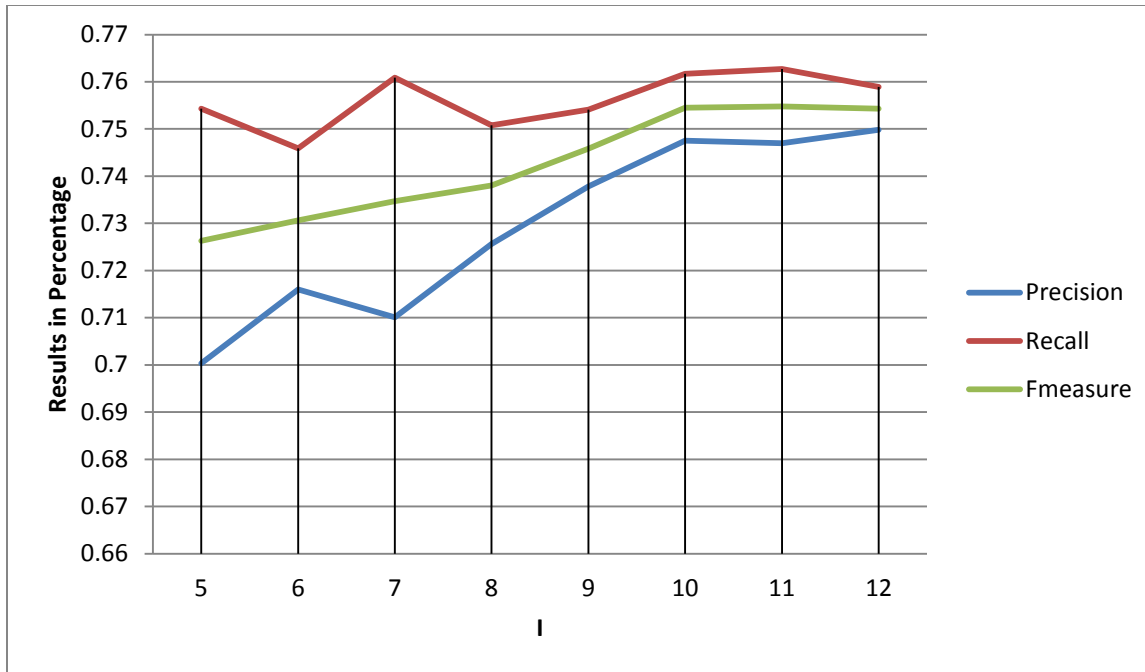
<b>K</b>	<b>Precision</b>	<b>Recall</b>	<b>Fmeasure</b>
5	0.2962	0.7698	0.4278
6	0.5255	0.8250	0.6420
7	0.5577	0.8085	0.6601
8	0.6091	0.8005	0.6918
9	0.6466	0.7874	0.7101
10	0.6615	0.7756	0.7140
11	0.6840	0.7747	0.7265
12	0.7402	0.7710	0.7553
13	0.7158	0.7851	0.7489
14	0.6905	0.7891	0.7365
15	0.6514	0.7929	0.7152

**Table 5.2.3.2 Evaluation Table for K=12,  
and Variations of I**

<b>I</b>	<b>Precision</b>	<b>Recall</b>	<b>Fmeasure</b>
5	0.70031	0.7543	0.7263
6	0.7160	0.7459	0.7306
7	0.7101	0.7609	0.7347
8	0.7256	0.7508	0.7380
9	0.7378	0.7541	0.7458
10	0.7475	0.7617	0.7545
11	0.7470	0.7627	0.7548
12	0.7498	0.7589	0.7543



**Figure 5.2.3.1 Graph of Evaluation Metrics with variations of K (19 august 2016)**



**Figure 5.2.3.2 Graph of Evaluation Metrics with variations of I (19 august 2016)**

**Table 5.2.3.3 Evaluation Table for I=12, and Variations of K**

<b>K</b>	<b>Precision</b>	<b>Recall</b>	<b>Fmeasure</b>
5	0.4392	0.8563	0.5794
6	0.4730	0.8473	0.6069
7	0.5705	0.8415	0.6787
8	0.5632	0.8221	0.6677
9	0.6098	0.8212	0.6984
10	0.6212	0.8023	0.6980
11	0.6417	0.7941	0.7085
12	0.6847	0.7879	0.7299
13	0.6780	0.7812	0.7251
14	0.6670	0.7728	0.7160
15	0.6069	0.7490	0.6705

**Table 5.2.3.4 Evaluation Table for K=12, and Variations of I**

<b>I</b>	<b>Precision</b>	<b>Recall</b>	<b>Fmeasure</b>
5	0.6782	0.7914	0.7298
6	0.6742	0.7812	0.7225
7	0.6984	0.7937	0.7420
8	0.6557	0.7870	0.7135
9	0.6853	0.7959	0.7348
10	0.6463	0.7789	0.7044
11	0.6481	0.7804	0.7071
12	0.6508	0.7821	0.7073

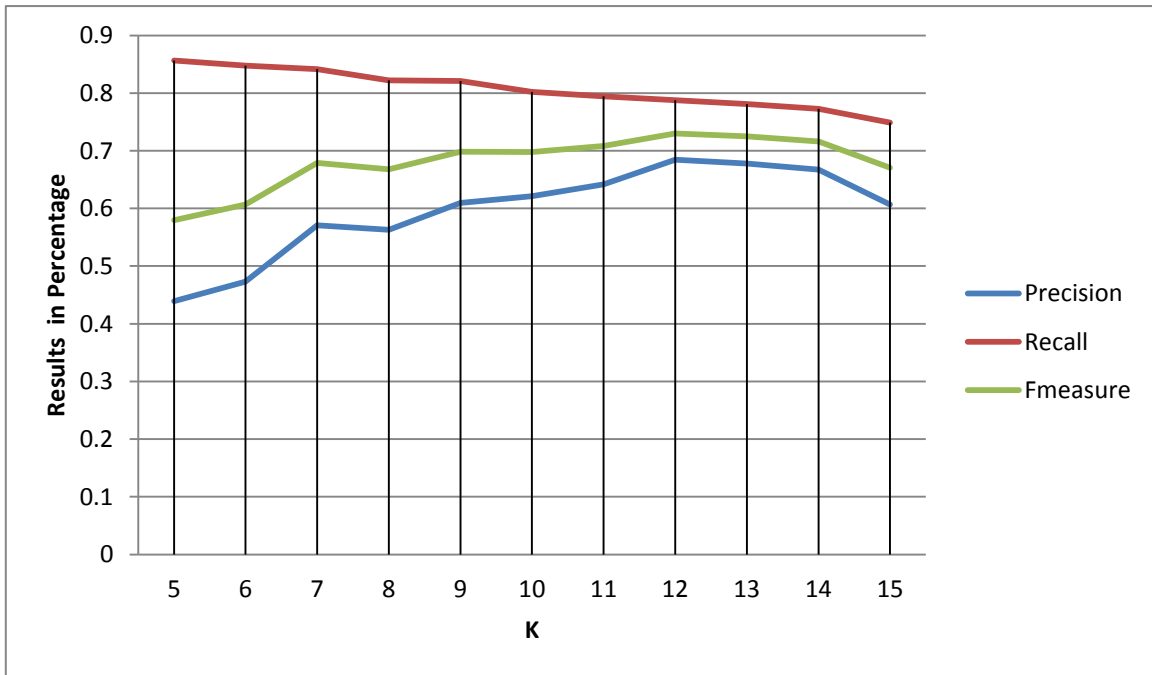


Figure. 5.2.3.3 Graph of Evaluation Metrics with variations of K (August 26, 2016)

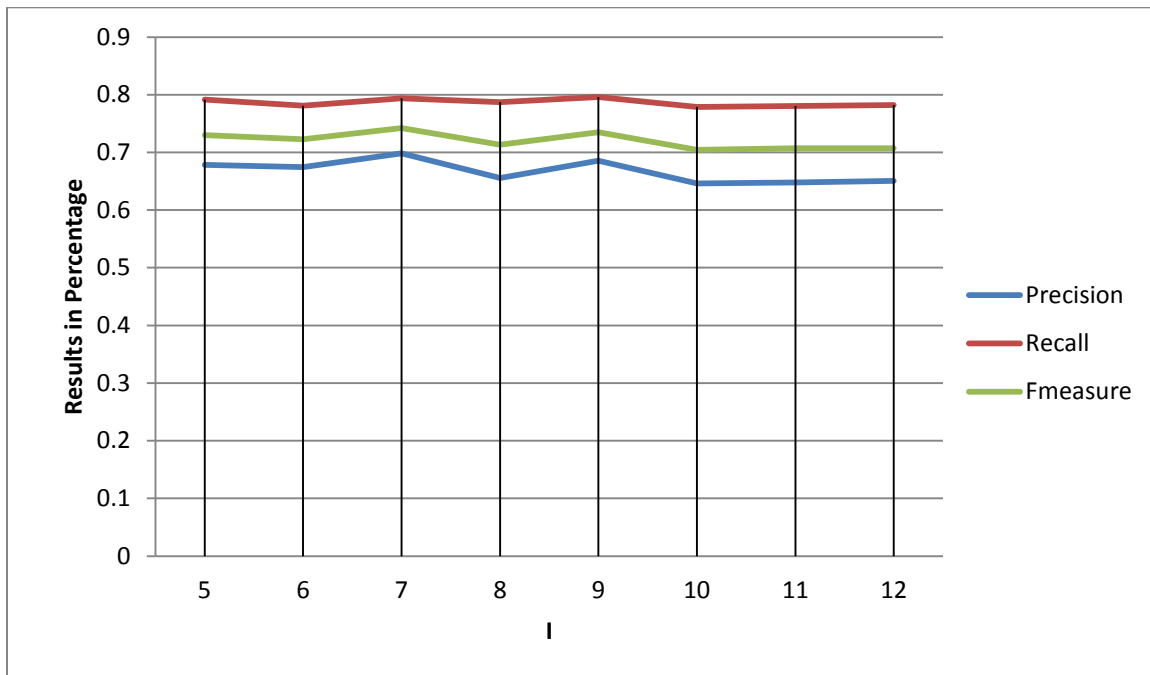


Figure. 5.2.3.4 Graph of Evaluation Metrics with variations of I (August 26, 2016)

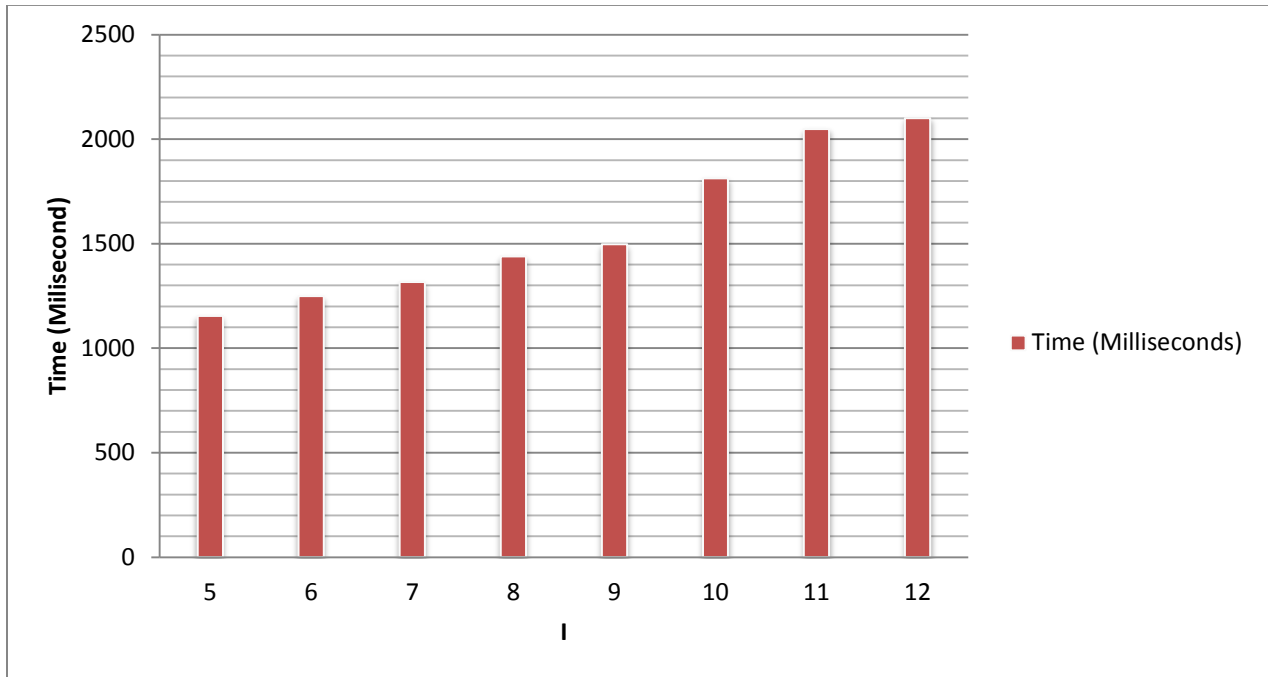
Time to complete the clustering process is tabulated as follows:

**Table 5.2.3.5. Completion Time of Clustering Process with different values of I (K=12)**

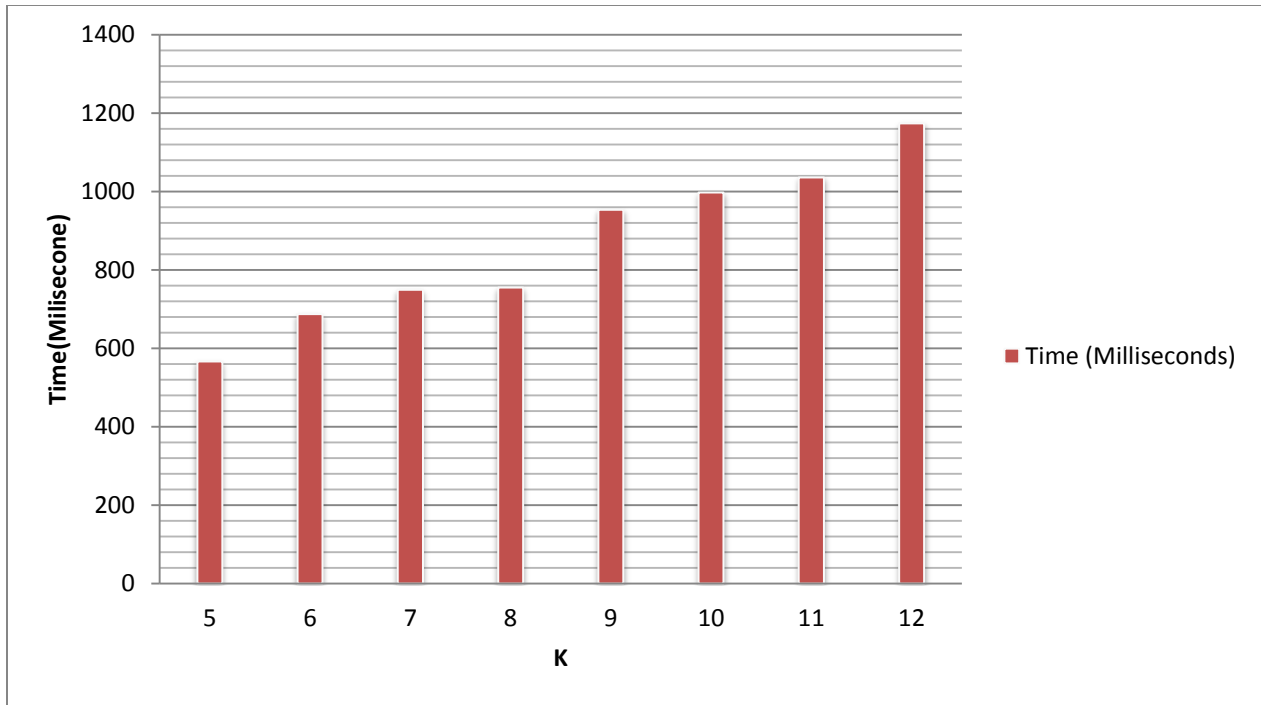
<b>I</b>	<b>Time (Milliseconds)</b>
5	1153.997447357
6	1248.188404786
7	1315.59471999
8	1438.740552225
9	1495.91359973
10	1812.084138415
11	2047.042212379
12	2100.17744123

**Table 5.2.3.6. Completion Time of Clustering Process with different values of K (I=6)**

<b>K</b>	<b>Time (Milliseconds)</b>
5	566.571551612
6	687.226066205
7	749.015990558
8	754.71924938
9	953.17702253
10	997.230870193
11	1035.973754392
12	1173.250517585



**Figure 5.2.3.5 Graph of Completion Time of Clustering Process with different values of I**



**Figure 5.2.3.6 Graph of Completion Time of Clustering Process with different values of K**

### 5.2.4 Result

From the experimental results, the result is efficient with the initial clusters seed 12 (K=12) and the iterations of no change in clusters centers depend upon the numbers of data sets. With the value of K=12; when the less number is data (638 News headings of 19 August, 2016), the value of f-measure is higher (75.53) and the F-measure is comparatively lower (72.99) when the number of data increased (791 News headings of 26 August, 2016).

When the less number of data (638 News headings of 19 August, 2016) then the less number of iterations (I=10) required and when the number of data increased (791 News headings of 26 August, 2016) then the less number of iterations (I=12) required to maintain the constant cluster canters in K-means clustering. As well as when the less number of data, the value of f-measure is higher (0.7545) and lower the f-measure (0.7277) when number of data increased.

While increasing the iteration (I), the completion time also increased. If initial cluster seed increased, completion time also increased. The time complexity of this work is the complexity of K-means clustering algorithm i.e.  $O(K)$ , where K is the initial clusters seed.

Therefore we can conclude that the number of iterations of changing the clusters mean of K-means clustering algorithm directly proportional to the number of data sets. The F-measure result is efficient with the value  $K=12$ . Running time is also directly proportional to the number of iterations and number of initial clusters seeds.



## CHAPTER 6

### CONCLUSION AND FUTURE WORK

#### 6.1 CONCLUSION

Only news headlines from the websites homepage are published. The headlines from other pages are not considered. Also, categorization of the news has not been done for the similar news found, i.e. the news headlines are not categorized into categories like national, business, etc. This work considers only those websites which are in English language. The online news portals which are designed in Nepali language are not considered. These portals may consist of similar news which is relevant with the news in the English publications. To include this part or feature, machine translation is required.

This work entitled “**News Clustering System based on Text Mining**” was developed for discovering or finding the similarities between the news articles extracted from different sources. Firstly, crawling and parsing methods for document retrieval from the web was applied. Then, the corpus of pre-processed document is prepared using various document pre-processing techniques. In the end, K-means clustering algorithm was implemented to discover the clusters of news items or articles and the efficiency of K-means algorithm was analyzed by Precision, Recall and F-measure evaluation metrics and finally the impact of initial value of K and iterations (I) and number of news (N) to discover the clusters were analyzed.

The similar news was grouped into a single cluster and presented in such a way that the news within the clusters was similar with each other. The real world application of this study is that it would help people to find detailed and more similar information about a particular news item in which the user is interested. This would not have been possible without the use of text mining techniques. In general, it is not feasible to manually look for similar news in each of the online news portals and then compare each of them to find similarities between them.

In the context of Nepal, not all online web newspapers are up-to-date. Hence, sometimes the clustering process might not be as accurate as expected.

## **6.2 FUTURE ENHANCEMENTS**

Some of the future enhancements that can be applied or done are as follows:

- Clustering of Nepali and English language can be done by using machine learning process (machine translation process).
- Categorization methods can be added to make the results and user interface more manageable and user friendly.
- Enhancement can be done to cover the news articles over large domains.
- The proposed method can be applied to other web documents such as research papers, digital libraries, etc.

## References

- [1] Azzopardi Joel, Staff Christopher, Faculty of ICT, University of Malta, Msida, *Incremental Clustering of News Reports*, An Article, ISSN 1999-4893, 24 August, 2012
- [2] Borglund Jon, Department of Information Technology, UPPSALA UNIVERSITET, *Event-Centric Clustering of News Articles*, A Thesis, October 2013.
- [3] Bun Khyou Khoo, Ishizuka Mitsuru, Department of Information and Communication Engineering, University of Tokyo, *Topic Extraction from News Archive Using TF-PDF Algorithm*.
- [4] Deshpande Rugved, Vaze Ketan, Rathod Suratsingh, Jarhad Tushar, *Comparative Study of Document Similarity Algorithms for Sentiment Analysis*, P.E.S Modern College of Engineering, Shivajinagar, Pune, India, September-October 2014.
- [5] Gomma H. Wael, Fahmy A. Aly, *A Survey of Text Similarity Approaches*, Cairo University, Cairo, Egypt, International Journal of Computer Applications (0975 – 8887) Volume 68– No.13, April 2013.
- [6] Hausler Christian, *Method for Determining the Similarity of Documents*, University of Applied Sciences and Arts, Northwestern Switzerland, 2012/13.
- [7] Huang Anna, *Similarity Measures for Text Document Clustering*, Department of Computer Science, The University of Waikato, New Zealand, April 2008.
- [8] Huang Zhen, Cardenas F. Alfonso, Computer Science Department, University of California, Los Angeles, *Extracting Hot Events from News Feeds, Visualization and Insights*,
- [9] Izzat Alsmadi, Zakaria Issa Saleh, IT faculty Yarmouk University Irbid, Jordan, *Documents Similarities Algorithms for Research Papers Authenticity*. ICCIT 2012.
- [10] K. Kalavendhan, P. Sumathi, Department of CSE, Tiruchengode, Namakkal, TamilNadu, *An Efficient Clustering Method To Find Similarity Between The Documents*, March 2014.

- [11] Khaled M.Hammouda. *Web Document Clustering Using Phrase-based Document Similarity*, 2002.
- [12] Lama Prabin, *CLUSTERING SYSTEM BASED ON TEXT MINING USING THE K-MEANS ALGORITHM*, TURKU UNIVERSITY OF APPLIED SCIENCES, 2013
- [13] Norvag Kjetil, Oyri Randi, *News Item Extraction for Text Mining in Web Newspapers*, Department of Computer and Information Science, Norwegian University of Science and Technology
- [14] Olowookere A. Toluwase(Department of Computer Science, University of Port Harcourt, Nigeria), Fasiku I. Ayodeji(Department of Computer Engineering, Ekiti State University, Nigeria), Emeto C. Ifeanyi(Department of Computer Science, University of Port Harcourt, Nigeria), *UPH Digital Library Miner: A Topic Modelling-based Software Application for Mining Document Collections of a Digital Library*, December 2015.
- [15] P. Nithya, R. Umamaheswari, Dr. N. Shanthi, *A Data Mining Objective Function with Feature Selection Algorithm using Document Clustering*, Gnanamani College of Technology, Namakkal, Tamilnadu, India, April 2015
- [16] Steinbach Michael, Karypis George, Kumar Vipin, *A Comparision of Document Clustering Techniques*, Department of Computer Science and Engineering, University of Minnesota.
- [17] Tung Thanh Khuat, Hung Duc Nguyen, Hanh Thi My Le, *A Comparision of Algorithms used to measure the similarity between two documents*, IJARCET, April 2015
- [18] TOMALA Karel, PLUCAR Jan, RAPANT Lukas, *The Data Extraction Using Distributed Crawler Inside the Multi-Agent System*, Technical University of Ostrava, Volume:11, December 2013.
- [19] Wael A. Gomma, Aly A. Fahmy, *A survey of Text Similarity Approaches*, Computer Science Department Faculty of Computers and Information, Cairo University Cairo, Egypt, April 2013.

[20] Zaho Ying, Karypis, Department of Computer Science, University of Minnesota, Minneapolis, *Comparison of Agglomerative and Partitional Document Clustering Algorithms*, 2008.

## **Bibliography**

- [http://en.wikipedia.org/wiki/Web\\_crawler](http://en.wikipedia.org/wiki/Web_crawler)
- <http://www.google.com/search?q=web+crawler+architecture&hl=en&prmd=imvns&tbm=isch&tbo=u&source=univ&sa=X&ei=n028T8CHNobSrQfk7IHcDQ&sqi=2&ved=0CGAQsAQ&biw=1366&bih=667>
- Ricardo Baeza-Yates, Berthier Ribeiro-Neto: *Morden Information Retrieval*
- <http://tfidf.com/>
- <http://www.celi.it/english/sophia.htm?referrer=google>
- <http://user.phil-fak.uni-duesseldorf.de/~rumpf/SS2003/Informationsextraktion/Pub/Eik99.pdf>
- <http://www.capital.health.vic.gov.au/capdev/PlanningEvaluation/FeasibilityStudy/FeasibilityStudyProcess/>

## APPENDIX

### Implementation Code

```
SOURCES = {  
  "The Himalayan Times" => "http://www.thehimalayantimes.com",  
  "Republica" => "http://www.myrepublica.com",  
  "The Kathmandu Post" => "http://kathmandupost.ekantipur.com",  
  "Nepal News" => "http://www.nepalnews.com",  
  "The Rising Nepal" => "http://therisingnepal.org.np/",  
  "Online Khabar" => "http://english.onlinekhabar.com/",  
  "NP News Portal" => "http://www.npnewsportal.com/",  
  "Nepali Headlines" => "http://nepaliheadlines.com/"  
}
```

```
SOURCES.each do |name, link|  
  puts "Creating news source: #{name}"  
  NewsSource.create(name: name, link: link)  
end
```

```
require 'open-uri'
```

```
namespace :crawler do  
  desc 'Gets news URLs'  
  task run: :environment do
```

```
    NewsSource.all.each do |source|  
      doc = Nokogiri::HTML(open(source.link))  
  
      links = doc.css('a').select { |l| l.content.length > 20 }
```

```
      puts "News Source: #{source.name}"  
      puts "Total Headings: #{links.count}"
```

```

links.each do |link|
  href = link.attributes["href"].value rescue next
  permalink = href =~ URI::regexp ? href : source.link + href
  source.news.find_or_create_by(title: link.content, link: permalink)
end
end

end

desc "Updates lemma"

task lemmatize: :environment do
  News.all.each do |news|
    news.lemma = news.lemmatize
    news.save
  end
end

end

desc "Save generated tokens based on News items"

task tokenize: :environment do
  Token.generate
end

end

namespace :token do
  namespace :tf do
    desc "Calculates TF"
    task calculate: :environment do

```

```

Token.all.each do |token|
  news_count = token.news_source.news.count
  token.tf = token.frequency.to_f / news_count.to_f
  token.save
end
end
end

```

```

namespace :idf do
  desc "Calculates IDF"
  task calculate: :environment do
    Token.all.each do |token|
      news_count = News.count
      global_token_frequency = Token.where(word: token.word).map(&:frequency).sum
      token.idf = Math.log10(news_count/global_token_frequency)
      token.save
    end
  end
end
end
end

```

```

class News < ActiveRecord::Base

```

```

  serialize :lemma, Array

```

```

STOP_WORDS = [
  "a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are", "arn't", "as", "at", "
  be", "because", "been", "before", "being", "below", "between", "both", "but", "by", "can", "can't", "cann
  ot", "could", "couldn't", "did", "didn't", "do", "does", "doesn't", "doing", "don't", "down", "during", "eac
  h", "few", "for", "from", "further", "had", "hadn't", "has", "hasn't", "have", "haven't", "having", "he", "he'
  d", "he'll", "he's", "her", "here", "here's", "hers", "herself", "him", "himself", "his", "how", "how's", "i", "i'

```



```
d", "i'll", "i'm", "i've", "if", "in", "into", "is", "isn't", "it", "it's", "its", "itself", "let's", "me", "more", "most", "mustn't", "my", "myself", "no", "nor", "not", "of", "off", "on", "once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shan't", "she", "she'd", "she'll", "she's", "uld", "shouldn't", "so", "some", "such", "than", "that", "that's", "the", "their", "theirs", "them", "themselves", "then", "there", "there's", "these", "they", "they'd", "they'll", "they're", "they've", "this", "those", "through", "to", "too", "under", "until", "up", "very", "was", "wasn't", "we", "we'd", "we'll", "we're", "we've", "were", "weren't", "what", "what's", "when", "when's", "where", "where's", "which", "while", "who", "who's", "whom", "why", "why's", "with", "won't", "would", "wouldn't", "you", "you'd", "you'll", "you're", "you've", "your", "yours", "yourself", "yourselves"]
```

```
belongs_to :news_source
```

```
belongs_to :cluster
```

```
has_many :tokens
```

```
def self.lemmatizer
```

```
  @@lemmatizer ||= Lemmatizer.new
```

```
end
```

```
def tokenize
```

```
  title.gsub(/[\^w\s]/, "").to_s.split(" ")
```

```
end
```

```
def without_stop_words
```

```
  tokenize - News::STOP_WORDS
```

```
end
```

```
def stem
```

```
  without_stop_words.map(&:stem)
```

```
end
```

```
def lemmatize
```

```

stem.map {|s| News.lemmatizer.lemma(s) }
end

def vector
  news_tokens = tokens.select {|t| t.news_source_id == news_source_id &&
lemma.map(&:downcase).include?(t.word) }
  Token.distinct.map do |t|
    token = news_tokens.select{ |l| l.word == t }.first
    token ? token.tf_idf : 0.0
  end
end

def similarity(news1)
  vector_1 = self.vector
  vector_2 = news1.vector
  product = vector_1.zip(vector_2).map {|p| p.map(&:to_f).inject(:*)}.compact.sum
  length_1 = Math.sqrt(vector_1.map{|i| i ** 2}.sum)
  length_2 = Math.sqrt(vector_2.map{|i| i ** 2}.sum)

  sim = product.to_f/(length_1*length_2).to_f
  sim.nan? ? 0.0 : sim
end

end

class Token < ActiveRecord::Base
  belongs_to :news_source
  belongs_to :news

  def self.generate
    News.all.each do |news|
      news.lemma.each do |lema|

```

```

    token = Token.find_by(word: lema.downcase, news_source: news.news_source, news:
news)
    if token
      token.increment!(:frequency)
    else
      Token.create(word: lema.downcase, news_source: news.news_source, frequency: 1, news:
news)
    end
  end
end
end
end

def self.distinct
  @@distinct ||= Token.pluck(:word).uniq
end

def tf_idf
  tf*idf
end

end

namespace :clusterify do
  desc "Clusterify"
  task run: :environment do

    # Random centers
    k = ENV['K'] || 12
    k = k.to_i
    i = ENV['I'] || 7
    i = i.to_i
    Cluster.delete_all

```

```

News.update_all(cluster_id: nil)

puts "Creating initial centers"
initial_centers = News.all.sample(k).to_a
k.times do |i|
  initial_center = initial_centers[i]
  cluster = Cluster.create(name: "Cluster_#{i+1}")
  initial_center.cluster = cluster
  initial_center.save
end
puts "Clustering items to the initial centers"
clusters = Cluster.all.to_a
news = News.all.includes(:tokens).to_a
(news - initial_centers).each do |n|
  rank = { }
  initial_centers.each do |initial_center|
    rank[initial_center.cluster_id] = n.similarity(initial_center)
  end
  highest_matching_key = rank.key(rank.values.max)
  unless rank[highest_matching_key].zero?
    n.cluster_id = highest_matching_key
  else
    n.cluster_id = clusters.sample.id
  end
  n.save
end

# Calculate new centers

i.times do |i|
  puts "Start of Iteration #{i}"

```

```

clusters.each do |cluster|
  cluster_news = cluster.news.reload.includes(:tokens)
  cluster_news_count = cluster_news.count
  cluster.mean = cluster_news.map(&:vector).transpose.map(&:sum).collect { |n|
n/cluster_news_count }
  puts "Cluster: #{cluster.name}"
  puts "Sum of mean: #{cluster.mean.sum}"
end

news.each do |n|
  rank = {}
  clusters.each do |c|
    rank[c.id] = n.similarity(c)
  end
  highest_matching_key = rank.key(rank.values.max)
  n.cluster_id = highest_matching_key
  n.save
end
puts "End of Iteration #{i}"
end

end
end
namespace :evaluation do
  desc "Calculating"
  task run: :environment do
    threshold = 0.12
    puts "Threshold: #{threshold}"
    Cluster.all.each do |cluster|
      puts "Calculating similarity between news of cluster: #{cluster.name}"

```

```

cluster_news = cluster.news.reload.includes(:tokens)
cluster_news_count = cluster_news.count
cluster.mean = cluster_news.map(&:vector).transpose.map(&:sum).collect { |n|
n/cluster_news_count }

similar_count = 0
cluster.news.each do |news|
  #puts "News: #{news.title}"
  #puts "Similarity: #{news.similarity(cluster)}"
  similar_count += 1 if news.similarity(cluster) > threshold
end

news_count = cluster.news.count
puts "Total news: #{news_count}"
puts "Similar news: #{similar_count}"

precision = similar_count.to_f/news_count.to_f
recall = similar_count.to_f/(similar_count + 10.0).to_f
cluster.update(precision: precision, recall: recall)
end
end
end
class Cluster < ActiveRecord::Base
  attr_accessor :mean
  has_many :news

  def agg_mean
    total_news = news.reload.includes(:tokens)
    total_news_count = total_news.count
    total_news.map(&:vector).transpose.map(&:sum).collect { |n| n/total_news_count }

```

end

def f\_measure

(2\*recall.to\_f\*precision.to\_f)/(precision.to\_f + recall.to\_f)

end

alias\_method :vector, :mean

end

## Evaluation Table

**20 August, 2016**

I	Precision	Recall	F-measure
5	0.7316	0.7641	0.7475
6	0.7394	0.7602	0.7497
7	0.7347	0.7619	0.7480
8	0.7504	0.7653	0.7578
9	0.7498	0.7671	0.7584
10	0.7544	0.7598	0.7571
11	0.7501	0.7567	0.7533
12	0.7506	0.7603	0.7554

K	Precision	Recall	F-measure
5	0.4791	0.8351	0.6089
6	0.5088	0.8192	0.6277
7	0.5496	0.8115	0.6554
8	0.6075	0.8001	0.6906
9	0.6044	0.8105	0.6924
10	0.6750	0.7875	0.7269
11	0.7000	0.7645	0.7309
12	0.7530	0.7639	0.7583
13	0.7544	0.7494	0.7518
14	0.7401	0.7414	0.7407
15	0.7200	0.7415	0.7306

**21 August, 2016**

I	Precision	Recall	F-measure
5	0.7047	0.7549	0.7289
6	0.7271	0.7640	0.7451
7	0.7524	0.7520	0.7522
8	0.7542	0.7576	0.7559
9	0.7592	0.7540	0.7566
10	0.7583	0.7634	0.7608
11	0.7585	0.7633	0.7608

K	Precision	Recall	F-measure
5	0.4659	0.8268	0.5952
6	0.5436	0.8206	0.6539
7	0.5539	0.8001	0.6547
8	0.6041	0.7981	0.6877
9	0.6502	0.7880	0.7125
10	0.6637	0.7757	0.7153
11	0.7128	0.7199	0.7163
12	0.7359	0.7539	0.7448
13	0.7270	0.7540	0.7402
14	0.7122	0.7329	0.7224



**22 August, 2016**

I	Precision	Recall	F-measure	K	Precision	Recall	F-measure
5	0.7142	0.7644	0.7385	5	0.4588	0.8353	0.5923
6	0.7223	0.7721	0.7464	6	0.4889	0.8144	0.6110
7	0.6409	0.7510	0.6916	7	0.5411	0.8074	0.6480
8	0.7289	0.7651	0.7465	8	0.5790	0.7939	0.6696
9	0.7491	0.7649	0.7570	9	0.6312	0.7867	0.7004
10	0.7525	0.7573	0.7548	10	0.6557	0.7764	0.7110
11	0.7395	0.7692	0.7541	11	0.6748	0.7707	0.7196
12	0.7394	0.7697	0.7543	12	0.7310	0.7593	0.7449
13	0.7457	0.7645	0.7550	13	0.7468	0.7161	0.7311
				14	0.7311	0.6854	0.7075

**23 August, 2016**

I	Precision	Recall	F-measure	K	Precision	Recall	F-measure
5	0.6393	0.7821	0.7036	5	0.4025	0.8483	0.5460
6	0.6911	0.7918	0.7380	6	0.4633	0.8402	0.5973
7	0.6865	0.7277	0.7065	7	0.5409	0.8371	0.6572
8	0.6806	0.7953	0.7335	8	0.5342	0.8181	0.6464
9	0.6533	0.7793	0.7108	9	0.5677	0.8075	0.6667
10	0.6562	0.7821	0.7137	10	0.5992	0.7997	0.6851
11	0.6598	0.7784	0.7142	11	0.6352	0.7923	0.7051
12	0.6660	0.7878	0.7218	12	0.6660	0.7882	0.7220
13	0.6650	0.7850	0.7200	13	0.6774	0.7709	0.7211
14	0.6693	0.79	0.7246				

**25 August, 2016**

I	Precision	Recall	F-measure	K	Precision	Recall	F-measure
5	0.6836	0.7969	0.7359	5	0.4411	0.8583	0.5805
6	0.6622	0.7855	0.7185	6	0.4815	0.8486	0.6137
7	0.6901	0.7936	0.7382	7	0.5016	0.8288	0.6222
8	0.6540	0.7836	0.7130	8	0.5515	0.8219	0.6586
9	0.6604	0.7848	0.7172	9	0.6045	0.8216	0.6933
10	0.6807	0.7926	0.7324	10	0.6105	0.8038	0.6925
11	0.6889	0.7828	0.7329	11	0.6432	0.8003	0.7098
12	0.6916	0.7858	0.7357	12	0.6444	0.78	0.7057